Two-Sided Derivatives for Regular Expressions and for Hairpin Expressions

Jean-Marc Champarnaud Jean-Philippe Dubernard
Hadrien Jeanne Ludovic Mignot

January 16, 2013

Abstract

The aim of this paper is to design the polynomial construction of a finite recognizer for hairpin completions of regular languages. This is achieved by considering completions as new expression operators and by applying derivation techniques to the associated extended expressions called hairpin expressions. More precisely, we extend partial derivation of regular expressions to two-sided partial derivation of hairpin expressions and we show how to deduce a recognizer for a hairpin expression from its two-sided derived term automaton, providing an alternative proof of the fact that hairpin completions of regular languages are linear context-free.

1 Introduction

The aim of this paper is to design the polynomial construction of a finite recognizer for hairpin completions of regular languages. Given an integer $k > 0$ and an involution H over an alphabet Γ, the hairpin k-completion of two languages L_1 and L_2 over Γ is the language $\mathrm{H}_k(L_1, L_2) = \{\alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \mid \alpha, \beta, \gamma \in \Gamma^* \wedge (\alpha\beta\gamma\mathrm{H}(\beta) \in L_1 \vee \beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \in L_2) \wedge |\beta| = k\}$ (see Figure 1). Hairpin completion has been deeply studied [2, 6, 9, 10, 11, 12, 13, 14, 16, 18, 19, 20, 21, 22, 23]. The hairpin completion of formal languages has been introduced in [9] by reason of its application to biochemistry. It aroused numerous studies that investigate theoretical and algorithmic properties of hairpin completions or related operations (see for example [14, 18, 21]). One of the most recent result concerns the problem of deciding regularity of hairpin completions of regular languages; it can be found in [11] as well as a complete bibliography about hairpin completion.

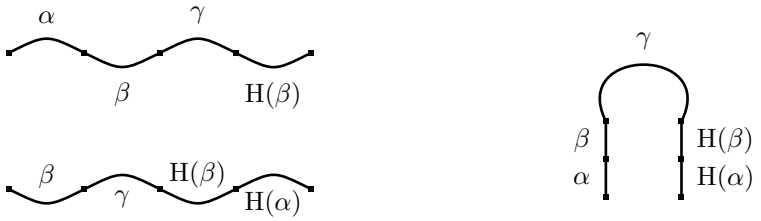

Figure 1: The Hairpin Completion.

Hairpin completions of regular languages are proved to be linear context-free from [9]. An alternative proof is presented in this paper, with a somehow more constructive approach, since it provides a recognizer for the hairpin completion. This is achieved by considering completions as new expression operators and by applying derivation techniques to the associated extended expressions, that we call hairpin expressions. Notice that a similar derivation-based approach has been used to study approximate regular expressions [8], through the definition of new distance operators.

Two-sided derivation is shown to be particularly suitable for the study of hairpin expressions. More precisely, we extend partial derivation of regular expressions [1] to two-sided partial derivation of regular expressions first and then of hairpin expressions. We prove that the set of two-sided derived terms of a hairpin expression E over an alphabet Γ is finite. Hence the two-sided derived term automaton A is a finite one. Furthermore the automaton A is over the alphabet $(\Gamma \cup \{\varepsilon\})^2$ and, as we prove it, the language over Γ of such an automaton is linear context-free and not necessarily regular. Finally we show that the language of the hairpin expression E and the language over Γ of the automaton A are equal.

This paper is an extended version of the conference paper [7]. It is organized as follows. Next section gathers useful definitions and properties concerning automata and regular expressions. The notion of two-sided residual of a language is introduced in Section 3, as well as the related notion of Γ-couple automaton. In Section 4, hairpin completions of regular languages and their two-sided residuals are investigated. The two-sided partial derivation of hairpin expressions is considered in Section 5, leading to the construction of a finite recognizer. A specific case is examined in Section 6.

2 Preliminaries

An *alphabet* is a finite set of distinct symbols. Given an alphabet Σ, we denote by Σ^* the set of all the words over Σ. The empty word is denoted by ε. A *language* over Σ is a subset of Σ^*. The three operations $\cup$, $\cdot$ and * are defined for any two languages L_1 and L_2 over Σ by: $L_1 \cup L_2 = \{w \in \Sigma^* \mid w \in L_1 \ \vee \ w \in L_2\}$, $L_1 \cdot L_2 = \{w_1 w_2 \in \Sigma^* \mid w_1 \in L_1 \ \wedge \ w_2 \in L_2\}$, $L_1^* = \{\varepsilon\} \cup \{w_1 \cdots w_k \in \Sigma^* \mid \forall j \in \{1, \ldots, k\}, w_j \in L_1\}$. The family of *regular languages* over Σ is the smallest

family $\mathcal{F}$ closed under the three operations $\cup$, $\cdot$ and * satisfying $\emptyset \in \mathcal{F}$ and $\forall a \in \Sigma,\ \{a\} \in \mathcal{F}$. Regular languages can be represented by *regular expressions*. A *regular expression* over Σ is inductively defined by: $E = a$, $E = \varepsilon$, $E = \emptyset$, $E = F + G$, $E = F \cdot G$, $E = F^*$, where a is any symbol in Σ and F and G are any two regular expressions over Σ. The *width of* E is the number of occurrences of symbols in E, and its *star number* the number of occurrences of the operator *. The language *denoted by* E is the language $L(E)$ inductively defined by: $L(A) = \{a\}$, $L(\varepsilon) = \{\varepsilon\}$, $L(\emptyset) = \emptyset$, $L(F + G) = L(F) \cup L(G)$, $L(F \cdot G) = L(F) \cdot L(G)$, $L(F^*) = (L(F))^*$, where a is any symbol in Σ and F and G are any two regular expressions over Σ. The language denoted by a regular expression is regular.

Let w be a word in Σ^* and L be a language. The *left residual* (resp. *right residual*) of L w.r.t. w is the language $w^{-1}(L) = \{w' \in \Sigma^* \mid ww' \in L\}$ (resp. $(L)w^{-1} = \{w' \in \Sigma^* \mid w'w \in L\}$). It has been shown that the set of the left residuals (resp. right residuals) of a language is a finite set if and only if the language is regular.

An *automaton* (or a *NFA*) over an alphabet Σ is a 5-tuple $A = (\Sigma, Q, I, F, \delta)$ where Σ is an alphabet, Q a finite set of *states*, $I \subset Q$ the set of *initial states*, $F \subset Q$ the set of *final states* and δ the *transition function* from $Q \times \Sigma$ to 2^Q. The domain of the function δ can be extended to $2^Q \times \Sigma^*$ as follows: for any word w in Σ^*, for any symbol a in Σ, for any set of states $P \subset Q$, for any state $p \in Q$, $\delta(P, \varepsilon) = P$, $\delta(p, aw) = \delta(\delta(p, a), w)$ and $\delta(P, w) = \bigcup_{p \in P} \delta(p, w)$.

The *language recognized* by the automaton A is the set $L(A) = \{w \in \Sigma^* \mid \delta(I, w) \cap F \neq \emptyset\}$. Given a state q in Q, the *right language* of q is the set $\overrightarrow{L}(q) = \{w \in \Sigma^* \mid \delta(q, w) \cap F \neq \emptyset\}$. It can be shown that **(1)** $L(A) = \bigcup_{i \in I} \overrightarrow{L}(i)$, **(2)** $\overrightarrow{L}(q) = \{\varepsilon \mid q \in F\} \cup (\bigcup_{a \in \Sigma, p \in \delta(q,a)} \{a\} \cdot \overrightarrow{L}(p))$ and **(3)** $a^{-1}(\overrightarrow{L}(q)) = \bigcup_{p \in \delta(q,a)} \overrightarrow{L}(p)$.

Kleene Theorem [15] asserts that a language is regular if and only if there exists an NFA that recognizes it. As a consequence, for any language L, there exists a regular expression E such that $L(E) = L$ if and only if there exists an NFA A such that $L(A) = L$. Conversion methods from an NFA to a regular expression and *vice versa* have been deeply studied. In this paper, we focus on the notion of partial derivative defined by Antimirov [1][1].

Given a regular expression E over an alphabet Σ and a word w in Σ^*, the *left partial derivative* of E w.r.t. w is the set $\frac{\partial}{\partial_w}(E)$ of regular expressions satisfying: $\bigcup_{E' \in \frac{\partial}{\partial_w}(E)} L(E') = w^{-1}(L(E))$.

This set is inductively computed as follows: for any two regular expressions F and G, for any word w in Σ^* and for any two distinct symbols a and b in Σ,

$$\frac{\partial}{\partial_a}(a) = \{\varepsilon\},\ \frac{\partial}{\partial_a}(b) = \frac{\partial}{\partial_a}(\varepsilon) = \frac{\partial}{\partial_a}(\emptyset) = \emptyset,$$
$$\frac{\partial}{\partial_a}(F + G) = \frac{\partial}{\partial_a}(F) \cup \frac{\partial}{\partial_a}(G),\ \frac{\partial}{\partial_a}(F^*) = \frac{\partial}{\partial_a}(F) \cdot F^*,$$
$$\frac{\partial}{\partial_a}(F \cdot G) = \begin{cases} \frac{\partial}{\partial_a}(F) \cdot G \cup \frac{\partial}{\partial_a}(G) & \text{if } \varepsilon \in L(F), \\ \frac{\partial}{\partial_a}(F) \cdot G & \text{otherwise,} \end{cases}$$

[1]Partial derivation is investigated in the more general framework of weighted expressions in [17].

$$\frac{\partial}{\partial_{aw}}(F) = \frac{\partial}{\partial_w}(\frac{\partial}{\partial_a}(F)),\ \frac{\partial}{\partial_\varepsilon}(F) = \{F\},$$

where for any set $\mathcal{E}$ of regular expressions, for any word w in Σ^*, for any regular expression F, $\frac{\partial}{\partial_w}(\mathcal{E}) = \bigcup_{E\in\mathcal{E}} \frac{\partial}{\partial_w}(E)$ and $\mathcal{E}\cdot F = \bigcup_{E\in\mathcal{E}}\{E\cdot F\}$. Any expression appearing in a left partial derivative is called a *left derived term*. Similarly, the *right partial derivative* of a regular expression E over an alphabet Σ w.r.t. a word w in Σ^* is the set $(E)\frac{\partial}{\partial_w}$ inductively defined as follows for any two regular expressions F and G, for any word w in Σ^* and for any two distinct symbols a and b in Σ,

$$(a)\frac{\partial}{\partial_a} = \{\varepsilon\},\ (b)\frac{\partial}{\partial_a} = (\varepsilon)\frac{\partial}{\partial_a} = (\emptyset)\frac{\partial}{\partial_a} = \emptyset,$$
$$(F+G)\frac{\partial}{\partial_a} = (F)\frac{\partial}{\partial_a} \cup (G)\frac{\partial}{\partial_a},\ (F^*)\frac{\partial}{\partial_a} = F^*\cdot(F)\frac{\partial}{\partial_a},$$
$$(F\cdot G)\frac{\partial}{\partial_a} = \begin{cases} F\cdot(G)\frac{\partial}{\partial_a} \cup (F)\frac{\partial}{\partial_a} & \text{if } \varepsilon\in L(G),\\ F\cdot(G)\frac{\partial}{\partial_a} & \text{otherwise,}\end{cases}$$
$$(F)\frac{\partial}{\partial_{aw}} = ((F)\frac{\partial}{\partial_a})\frac{\partial}{\partial_w},\ (F)\frac{\partial}{\partial_\varepsilon} = \{F\},$$

where for any set $\mathcal{E}$ of regular expressions, for any word w in Σ^*, for any regular expression F, $(\mathcal{E})\frac{\partial}{\partial_w} = \bigcup_{E\in\mathcal{E}}(E)\frac{\partial}{\partial_w}$ and $F\cdot\mathcal{E} = \bigcup_{E\in\mathcal{E}}\{F\cdot E\}$. Any expression appearing in a right partial derivative is called a *right derived term*. We denote by $\overleftarrow{\mathcal{D}_E}$ (resp. $\overrightarrow{\mathcal{D}_E}$) the set of left (resp. right) derived terms of the expression E. From the set of left derived terms of a regular expression E of width n, Antimirov defined in [1] the *derived term automaton* A of E and showed that A is a k-state NFA that recognizes $L(E)$, with $k \leq n+1$.

A language over an alphabet Γ is said to be *linear context-free* if it can be generated by a linear grammar, that is a grammar equipped with productions in one of the following forms:

1. $A \to xBy$, where A and B are any two non-terminal symbols, and x and y are any two symbols in $\Gamma\cup\{\varepsilon\}$ such that $(x,y)\neq(\varepsilon,\varepsilon)$,
2. $A\to\varepsilon$, where A is any non-terminal symbol.

Notice that the family of regular languages is strictly included into the family of linear context-free languages. In the following, we will consider combinations of left and right partial derivatives in order to deal with non-regular languages.

3 Two-sided Residuals of a Language and Couple NFA

In this section, we extend residuals to two-sided residuals. This operation is the composition of left and right residuals, but it is more powerful than classical residuals since it allows to compute a finite subset of the set of residuals even for non-regular languages, which leads to the construction of a derivative-based finite recognizer.

Definition 1. *Let L be a language over an alphabet Γ and let u and v be two words in Γ^*. The* two-sided residual *of L w.r.t. (u,v) is the language $(u,v)^{-1}(L) = \{w\in\Gamma^* \mid uwv\in L\}$.*

As above-mentioned, the two-sided residual operation is the composition of the two operations of left and right residuals.

Lemma 1. *Let L be a language over an alphabet Γ and u and v be two words in Γ^*. Then: $(u,v)^{-1}(L) = (u^{-1}(L))v^{-1} = u^{-1}((L)v^{-1})$.*

Proof. Let w be a word in Γ^*.

$$\begin{array}{lll} w \in (u^{-1}(L))v^{-1} \Leftrightarrow wv \in u^{-1}(L) \Leftrightarrow uwv \in L & \Leftrightarrow (u,v)^{-1}(L) \\ \Leftrightarrow uwv \in L & \Leftrightarrow uw \in (L)v^{-1} \Leftrightarrow w \in u^{-1}((L)v^{-1}). \end{array}$$

□

Corollary 1. *Let L be a language over an alphabet Γ and u and v be two words in Γ^*. Then: $\varepsilon \in (u,v)^{-1}(L) \Leftrightarrow uv \in L$.*

It is a folk knowledge that NFAs are related to left residual computation according to the following assertion **(A)**: in an NFA $(\Sigma, Q, I, F, \delta)$, a word aw belongs to $\overrightarrow{L}(q)$ with $q \in Q$ if and only if w belongs to $a^{-1}(\overrightarrow{L}(q)) = \bigcup_{q' \in \delta(q,a)} \overrightarrow{L}(q')$. Since a two-sided residual w.r.t. a couple (x,y) of symbols in an alphabet Γ is by definition the combination of a left residual w.r.t. x and of a right residual w.r.t. y, the assertion **(A)** can be extended to two-sided residuals by introducing *couple NFAs* equipped with transitions labelled by couples of symbols in Γ. The notion of right language of a state is extended to the one of Γ-right language as follows: if a given word w in Γ^* belongs to the Γ-right language of a state q' and if there exists a transition from a state q to q' labelled by a couple (x,y), then the word xwy belongs to the Γ-right language of q.

More precisely, given an alphabet Γ, we set $\Sigma_\Gamma = \{(x,y) \mid x,y \in \Gamma \cup \{\varepsilon\} \wedge (x,y) \neq (\varepsilon,\varepsilon)\}$. We consider the mapping Im from $(\Sigma_\Gamma)^*$ to Γ^* inductively defined for any word w in $(\Sigma_\Gamma)^*$ and for any symbol $(x,y) \in \Sigma_\Gamma$ by: $\mathrm{Im}(\varepsilon) = \varepsilon$ and $\mathrm{Im}((x,y) \cdot w) = x \cdot \mathrm{Im}(w) \cdot y$. Notice that this mapping was introduced by Sempere [24] in order to compute the language denoted by a *linear expression*. Linear expressions denote linear context-free languages, and are equivalent to the *regular-like expressions* of Brzozowski [3].

Definition 2. *Let $A = (\Sigma, Q, I, F, \delta)$ be an NFA. The NFA A is a* couple NFA *if there exists an alphabet Γ such that $\Sigma \subset \Sigma_\Gamma$. In this case, A is called a* Γ-couple NFA. *The* Γ-language *of a Γ-couple NFA A is the subset $L_\Gamma(A)$ of Γ^* defined by: $L_\Gamma(A) = \{\mathrm{Im}(w) \mid w \in L(A)\}$.*

The definition of right languages and their classical properties extend to couple NFAs as follows. Let $A = (\Sigma, Q, I, F, \delta)$ be a Γ-couple NFA and q be a state in Q. The Γ-*right language* of q is the subset $\overrightarrow{L}_\Gamma(q)$ of Γ^* defined by: $\overrightarrow{L}_\Gamma(q) = \{\mathrm{Im}(w) \mid w \in \overrightarrow{L}(q)\}$.

Lemma 2. *Let $A = (\Sigma, Q, I, F, \delta)$ be a Γ-couple NFA and q be a state in Q. Then: $L_\Gamma(A) = \bigcup_{i \in I} \overrightarrow{L}_\Gamma(i)$.*

Proof. Trivially deduced from Definition 2, from definition of Γ-right languages and from the fact that $L(A) = \bigcup_{i \in I} \overrightarrow{L}(i)$. □

Lemma 3. *Let $A = (\Sigma, Q, I, F, \delta)$ be a Γ-couple NFA and q be a state in Q. Then:* $\overrightarrow{L}_\Gamma(q) = \{\varepsilon \mid q \in F\} \cup \bigcup_{(x,y)\in\Sigma, q'\in\delta(q,(x,y))} \{x\} \cdot \overrightarrow{L}_\Gamma(q') \cdot \{y\}$.

Proof. Trivially deduced from Definition 2, from definition of Γ-right languages and from the fact that $\overrightarrow{L}(q) = \{\varepsilon \mid q \in F\} \cup \bigcup_{a\in\Sigma, q'\in\delta(q,a)} \{a\} \cdot \overrightarrow{L}(q')$. □

Corollary 2. *Let $A = (\Sigma, Q, I, F, \delta)$ be a Γ-couple NFA, (x, y) be a couple in Σ_Γ and q be a state in Q. Then:* $(x, y)^{-1}(\overrightarrow{L}_\Gamma(q)) = \bigcup_{q'\in\delta(q,(x,y))} \overrightarrow{L}_\Gamma(q')$.

The following example illustrates the fact that there exist non-regular languages that can be recognized by couple NFAs.

Example 1. *Let $\Gamma = \{a, b\}$ and A be the automaton of the Figure 2. The Γ-language of A is $L_\Gamma(A) = \{a^n b^n \mid n \in \mathbb{N}\}$.*

Figure 2: The Couple Automaton A.

As a consequence there exist non-regular languages that are recognized by a couple NFA. In fact, the family of languages recognized by couple NFAs is exactly the family of linear context-free languages.

Proposition 1. *The Γ-language recognized by a Γ-couple NFA is linear context-free.*

Proof. Let $A = (\Sigma, Q, I, F, \delta)$. Let us define the grammar $G = (X, V, P, S)$ by:

- $X = \Gamma$, the set of terminal symbols,
- $V = \{A_q \mid q \in Q\} \cup \{S\}$, the set of non-terminal symbols,
- $P = \{S \to A_q \mid q \in I\} \cup \{A_q \to \varepsilon \mid q \in F\} \cup \{A_q \to \alpha A_{q'} \beta \mid q' \in \delta(q, (\alpha, \beta))\}$, the set of productions,
- S, the axiom.

1. Let w be word in Γ^*. Let us first show that w belongs to the language generated by the grammar $G_q = (X, V, P, A_q)$ if and only if it is in $\overrightarrow{L}_\Gamma(q)$, by recurrence over the length of w.
   (a) Let us suppose that $w = \varepsilon$. By construction of G_q, $A_q \to \varepsilon$ if and only if $q \in F$, *i.e.* $\varepsilon \in \overrightarrow{L}_\Gamma(q)$.
   (b) Let us suppose that $w = \alpha w' \beta$ with $(\alpha, \beta) \neq (\varepsilon, \varepsilon)$. By definition of $L(G_q)$, $w \in L(G_q)$ if there exists a symbol $A_{q'}$ in V such that $A_q \to \alpha A_{q'} \beta$ and $w' \in L(G_{q'})$. By recurrence hypothesis, it holds

that $w' \in L(G_{q'}) \Leftrightarrow w' \in \overrightarrow{L}_\Gamma(q')$. Since by construction $A_q \to \alpha A_{q'} \beta \Leftrightarrow q' \in \delta(q,(\alpha,\beta))$ and since according to Lemma 3, $\overrightarrow{L}_\Gamma(q) = \{\varepsilon \mid q \in F\} \cup \bigcup_{(x,y)\in\Sigma, q'\in\delta(q,(x,y))} \{x\} \cdot \overrightarrow{L}_\Gamma(q') \cdot \{y\}$, it holds that $w \in L(G_q) \Leftrightarrow w \in \overrightarrow{L}_\Gamma(q)$.

2. Since $L(G) = \bigcup_{q|S\to A_q} L(G_q)$, it holds from **(1)** that $L(G) = \bigcup_{q\in I} \overrightarrow{L}_\Gamma(q)$, that equals according to Lemma 2 to $L(A)$.

Finally, since the Γ-language of A is generated by a linear grammar, it is linear context free. □

Proposition 2. *The language generated by a linear grammar is recognized by a couple NFA.*

Proof. Let $G = (X, V, P, S)$ be a linear grammar. Let us define the automaton $A = (\Sigma, Q, I, F, \delta)$ by:

- $\Sigma = \Sigma_X$,
- $Q = V$,
- $I = \{S\}$,
- $F = \{B \in V \mid (B \to \varepsilon) \in P\}$,
- $B' \in \delta(B,(x,y)) \Leftrightarrow (B \to xBy) \in P$.

For any symbol B in V, let us set $G_B = (X, V, P, B)$. Let w be a word in X^*. Let us show by recurrence over the length of w that $w \in L(G_B) \Leftrightarrow w \in \overrightarrow{L}_X(B)$.

1. Let $w = \varepsilon$. Then $\varepsilon \in L(G_B)$ if and only if $(G_B \to \varepsilon) \in P$. By construction, it is equivalent to $B \in F$ and to $\varepsilon \in \overrightarrow{L}_X(B)$.

2. Let us suppose that w is different from ε. Then by recurrence hypothesis and according to Lemma 3:

$w \in L(G_B)$
$\Leftrightarrow \exists (x,y) \in \Sigma_X, w' \in X^*, B' \in V \mid w = xw'y \wedge (B \to xB'y) \in P \wedge w' \in L(G_{B'})$
$\Leftrightarrow \exists (x,y) \in \Sigma_X, w' \in X^*, B' \in V \mid w = xw'y \wedge B' \in \delta(B,(x,y)) \wedge w' \in \overrightarrow{L}_X(B')$
$\Leftrightarrow w \in \overrightarrow{L}_X(B)$

Finally, since $L(G) = L(G_S) = \overrightarrow{L}_S(B)$, it holds from Lemma 2 that $L(G) = L(A)$. □

Theorem 1. *A language is linear context-free if and only if it is recognized by a couple NFA.*

Proof. Directly from Proposition 1 and from Proposition 2. □

We present here two algorithms in order to solve the membership problem[2] *via* a couple NFA. The Algorithm 2 checks whether the word $w \in \Gamma^*$ is recognized by the Γ-couple NFA A. It returns TRUE if there exists an initial state such that its Γ-right language contains w. The Algorithm 1 checks whether the word $w \in \Gamma^*$ is in the Γ-right language of the state q.

Algorithm 1 IsInRightLanguage(A,w,q)

Require: $A = (\Sigma, Q, I, F, \delta)$ a Γ-couple NFA, w a word in Γ^*, q a state in Q
Ensure: Returns $(w \in \overrightarrow{L}_\Gamma(q))$

```
1: if w = ε then
2:    P ← (q ∈ F)
3: else
4:    P ← FALSE
5:    for all (q, (α, β), q') ∈ δ | w = αw'β do
6:       P ← P ∨ IsInRightLanguage(A, w', q')
7:    end for
8: end if
9: return P
```

Algorithm 2 MembershipTest(A,w)

Require: $A = (\Sigma, Q, I, F, \delta)$ a Γ-couple NFA, w a word in Γ^*
Ensure: Returns $(w \in L_\Gamma(A))$

```
1: R ← FALSE
2: for all i ∈ I do
3:    R ← R ∨ IsInRightLanguage(A, w, i)
4: end for
5: return R
```

Proposition 3. *Let $A = (\Sigma, Q, I, F, \delta)$ be a Γ-couple NFA, q be a state in Q and w be a word in Γ^*. The two following propositions are satisfied:*

1. *Algorithm 1: IsInRightLanguage(A, w, q) returns $(w \in \overrightarrow{L}_\Gamma(q))$,*
2. *Algorithm 2: MembershipTest(A,w) returns $(w \in L_\Gamma(A))$.*

Proof. Let w be a word in Γ^*.

1. Let us show by recurrence over the length of w that the algorithm IsInRightLanguage(A, w, q) returns $(w \in \overrightarrow{L}_\Gamma(q))$.

   If $w = \varepsilon$, $P = TRUE \Leftrightarrow q \in F \Leftrightarrow \varepsilon \in \overrightarrow{L}_\Gamma(q)$.

   Let us suppose now that $|w| \geq 1$. Then $P = \bigvee_{(q,(\alpha,\beta),q') \in \delta | w = \alpha w' \beta}$ IsInRightLanguage(A, w', q'). If there is no transition $(q, (\alpha, \beta), q') \in \delta$,

[2] Given a language L and a word w, does w belong to L?

then trivially $w \notin \overrightarrow{L}_\Gamma(q)$. For any $(q, (\alpha, \beta), q') \in \delta$, let us notice that $(\alpha, \beta) \in \Sigma_\Gamma$. As a consequence, the length of any word w' satisfying $w = \alpha w' \beta \ \wedge \ (q, (\alpha, \beta), q') \in \delta$ is strictly smaller than $|w|$. Let w' be a word satisfying $w = \alpha w' \beta \ \wedge \ (q, (\alpha, \beta), q') \in \delta$. According to recurrence hypothesis, IsInRightLanguage(A, w', q') returns $(w' \in \overrightarrow{L}_\Gamma(q'))$. Hence $P = \bigvee_{(q,(\alpha,\beta),q') \in \delta | w = \alpha w' \beta} (w' \in \overrightarrow{L}_\Gamma(q'))$. Finally, according to Lemma 3, $P = (w \in \overrightarrow{L}_\Gamma(q))$.

2. Since $R = \bigvee_{i \in I}$ IsInRightLanguage(A, w, i), it holds as a direct consequence that $R = \bigvee_{i \in I} (w \in \overrightarrow{L}_\Gamma(i))$. Hence, according to Lemma 2, it holds that $R = (w \in L_\Gamma(A))$.

□

The following sections are devoted to hairpin completions and their two-sided residuals. It turns out that hairpin completions are linear context-free languages. Hence, we show how to compute a couple NFA that recognizes a given hairpin completion.

4 Hairpin Completion of a Language and its Residuals

Let Γ be an alphabet. An *involution* f over Γ is a mapping from Γ to Γ satisfying for any symbol a in Γ, $\mathrm{f}(\mathrm{f}(a)) = a$. An *anti-morphism* μ over Γ^* is a mapping from Γ^* to Γ^* satisfying for any two words u and v in Γ^* $\mu(u \cdot v) = \mu(v) \cdot \mu(u)$. Any mapping g from Γ to Γ can be extended as an anti-morphism over Γ^* as follows: $\forall a \in \Gamma$, $\forall w \in \Gamma^*$, $\mathrm{g}(\varepsilon) = \varepsilon$, $\mathrm{g}(a \cdot w) = \mathrm{g}(w) \cdot \mathrm{g}(a)$.

Definition 3. *Let Γ be an alphabet and* H *be an anti-morphism over Γ^*. Let L_1 and L_2 be two languages over Γ. Let $k > 0$ be an integer. The* (H, k)-completion *of L_1 and L_2 is the language $\mathrm{H}_k(L_1, L_2)$ defined by:*

$$\mathrm{H}_k(L_1, L_2) = \{\alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \mid \alpha, \beta, \gamma \in \Gamma^* \wedge (\alpha\beta\gamma\mathrm{H}(\beta) \in L_1 \vee \beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \in L_2) \wedge |\beta| = k\}.$$

The (H, k)-completion operator can be defined as the union of two unary operators $\overleftarrow{\mathrm{H}_k}$ and $\overrightarrow{\mathrm{H}_k}$.

Definition 4. *Let Γ be an alphabet and* H *be an anti-morphism over Γ^*. Let L be a language over Γ. Let $k > 0$ be an integer. The right (resp. left)* (H, k)-completion *of L is the language $\overrightarrow{\mathrm{H}_k}(L)$ (resp. $\overleftarrow{\mathrm{H}_k}(L)$) defined by:*

$$\overrightarrow{\mathrm{H}_k}(L) = \{\alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \mid \alpha, \beta, \gamma \in \Gamma^* \ \wedge \ \alpha\beta\gamma\mathrm{H}(\beta) \in L \wedge |\beta| = k\},$$
$$\overleftarrow{\mathrm{H}_k}(L) = \{\alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \mid \alpha, \beta, \gamma \in \Gamma^* \ \wedge \ \beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \in L \wedge |\beta| = k\}.$$

Lemma 4. *Let Γ be an alphabet and* H *be an anti-morphism over Γ^*. Let L_1 and L_2 be two languages over Γ. Let $k > 0$ be an integer. Then:*

$$\mathrm{H}_k(L_1, L_2) = \overrightarrow{\mathrm{H}_k}(L_1) \cup \overleftarrow{\mathrm{H}_k}(L_2).$$

Proof. Let w be a word in Γ^*.

$$w \in \mathrm{H}_k(L_1, L_2) \Leftrightarrow \begin{cases} w = \alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \\ \wedge(\alpha\beta\gamma\mathrm{H}(\beta) \in L_1 \vee \beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \in L_2) \\ \wedge|\beta| = k \end{cases}$$
$$\Leftrightarrow \begin{cases} (w = \alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \wedge \alpha\beta\gamma\mathrm{H}(\beta) \in L_1 \wedge\ |\beta| = k) \\ \vee(w = \alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \wedge \beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \in L_2 \wedge\ |\beta| = k) \end{cases}$$
$$\Leftrightarrow w \in \overrightarrow{\mathrm{H}_k}(L_1) \vee w \in \overleftarrow{\mathrm{H}_k}(L_2) \Leftrightarrow w \in \overrightarrow{\mathrm{H}_k}(L_1) \cup \overleftarrow{\mathrm{H}_k}(L_2).$$

□

When H is an involution over Γ, the (H, k)-completion of L_1 and L_2 is called a hairpin completion [9]. Even in the case where H is not an involution, we will say that languages such as $\overrightarrow{\mathrm{H}_k}(L)$, $\overleftarrow{\mathrm{H}_k}(L)$ or $\mathrm{H}_k(L, L')$ are hairpin completed languages and we will speak of hairpin completions. We first establish formulae in this general setting in order to compute the two-sided residuals of the completed language of an arbitrary language. The following operator is useful.

Definition 5. *Let Γ be an alphabet and* H *be an anti-morphism over Γ^*. Let L be a language over an alphabet Γ. Let $k > 0$ be an integer. The language $\mathrm{H}'_k(L)$ is defined by:* $\mathrm{H}'_k(L) = \{\beta\gamma\mathrm{H}(\beta) \in L \mid \beta, \gamma \in \Gamma^* \ \wedge\ |\beta| = k\}$.

We split the computation of two-sided residuals of a completed language w.r.t. (x, y) couples: the first case is when both x and y are symbols.

Lemma 5. *Let Γ be an alphabet and* H *be an anti-morphism over Γ^*. Let L be a language over an alphabet Γ. Let $k > 0$ be an integer. Let L' be a language in $\{\overleftarrow{\mathrm{H}_k}(L), \overrightarrow{\mathrm{H}_k}(L), \mathrm{H}'_k(L)\}$. Let w a word in Γ^*. Then:*

$$w \in L' \Rightarrow |w| \geq k \ \wedge\ \exists a \in \Gamma, \exists w' \in \Gamma^*, w = aw'\mathrm{H}(a).$$

Proof. Trivially deduced from Definition 4 and Definition 5. □

Corollary 3. *Let Γ be an alphabet and* H *be an anti-morphism over Γ^*. Let L be a language over an alphabet Γ. Let $k > 0$ be an integer. Let L' be a language in $\{\overleftarrow{\mathrm{H}_k}(L), \overrightarrow{\mathrm{H}_k}(L), \mathrm{H}'_k(L)\}$. Then: $L' = \bigcup_{x\in\Gamma}\{x\} \cdot ((x, \mathrm{H}(x))^{-1}(L')) \cdot \{\mathrm{H}(x)\}$.*

Proposition 4. *Let Γ be an alphabet and* H *be an anti-morphism over Γ^*. Let L be a language over Γ. Let (x, y) a couple of symbols in $\Gamma \times \Gamma$. Let $k > 0$ be an integer. Then:*

$$(x,y)^{-1}(\overrightarrow{\mathrm{H}_k}(L)) = \begin{cases} \emptyset & \text{if } y \neq \mathrm{H}(x), \\ \overrightarrow{\mathrm{H}_k}(x^{-1}(L)) \cup (x,y)^{-1}(L) & \text{if } y = \mathrm{H}(x) \ \wedge\ k = 1, \\ \overrightarrow{\mathrm{H}_k}(x^{-1}(L)) \cup \mathrm{H}'_{k-1}((x,y)^{-1}(L)) & \text{otherwise,} \end{cases}$$

$$(x,y)^{-1}(\overleftarrow{\mathrm{H}_k}(L)) = \begin{cases} \emptyset & \text{if } y \neq \mathrm{H}(x), \\ \overleftarrow{\mathrm{H}_k}((L)y^{-1}) \cup (x,y)^{-1}(L) & \text{if } y = \mathrm{H}(x) \ \wedge\ k = 1, \\ \overleftarrow{\mathrm{H}_k}((L)y^{-1}) \cup \mathrm{H}'_{k-1}((x,y)^{-1}(L)) & \text{otherwise,} \end{cases}$$

$$(x,y)^{-1}(\mathrm{H}'_k(L)) = \begin{cases} \emptyset & \text{if } y \neq \mathrm{H}(x), \\ \mathrm{H}'_{k-1}((x,y)^{-1}(L)) & \text{if } y = \mathrm{H}(x) \ \wedge\ k > 1, \\ (x,y)^{-1}(L) & \text{otherwise.} \end{cases}$$

Proof. Let w be a word in Γ^*. According to Lemma 5, any word u in $\overrightarrow{\mathrm{H}_k}(L) \cup \overleftarrow{\mathrm{H}_k}(x^{-1}(L)) \cup \mathrm{H}'_k(L)$ can be split up into avb with $b = \mathrm{H}(a)$. As a consequence, whenever $y \neq \mathrm{H}(x)$, it holds that $(x,y)^{-1}(\overrightarrow{\mathrm{H}_k}(L)) = (x,y)^{-1}(\overleftarrow{\mathrm{H}_k}(L)) = (x,y)^{-1}(\mathrm{H}'_k(L)) = \emptyset$. Let us suppose now that $y = \mathrm{H}(x)$.

(I) Let us define the languages L_1 and L_2 by:

$$L_1 = (x,y)^{-1}(\overrightarrow{\mathrm{H}_k}(L)),$$
$$L_2 = \begin{cases} \overrightarrow{\mathrm{H}_k}(x^{-1}(L)) \cup \mathrm{H}'_{k-1}((x,y)^{-1}(L)) & \text{if } k > 1, \\ \overrightarrow{\mathrm{H}_k}(x^{-1}(L)) \cup (x,y)^{-1}(L) & \text{otherwise.} \end{cases}$$

Then:

$$\begin{aligned} w \in L_1 &\Leftrightarrow xwy \in \overrightarrow{\mathrm{H}_k}(L) \\ &\Leftrightarrow \begin{cases} (xwy = x\alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha)y \wedge y = \mathrm{H}(x) \wedge x\alpha\beta\gamma\mathrm{H}(\beta) \in L \wedge |\beta| = k) \\ \vee (xwy = x\beta\gamma\mathrm{H}(\beta)y \wedge y = \mathrm{H}(x) \wedge x\beta\gamma\mathrm{H}(\beta)y \in L \wedge |\beta| = k-1) \end{cases} \\ &\Leftrightarrow \begin{cases} (w = \alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \wedge y = \mathrm{H}(x) \wedge \alpha\beta\gamma\mathrm{H}(\beta) \in x^{-1}(L) \wedge |\beta| = k) \\ \vee (w = \beta\gamma\mathrm{H}(\beta) \wedge y = \mathrm{H}(x) \wedge \beta\gamma\mathrm{H}(\beta) \in (x,y)^{-1}(L) \wedge |\beta| = k-1) \end{cases} \\ &\Leftrightarrow \begin{cases} (w = \alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \wedge y = \mathrm{H}(x) \wedge w \in \overrightarrow{\mathrm{H}_k}(x^{-1}(L))) \\ \vee (w = \beta\gamma\mathrm{H}(\beta) \wedge y = \mathrm{H}(x) \wedge w \in \mathrm{H}'_{k-1}((x,y)^{-1}(L)) \wedge k \neq 1) \\ \vee (w = \gamma \wedge y = \mathrm{H}(x) \wedge w \in (x,y)^{-1}(L) \wedge k = 1) \end{cases} \\ &\Leftrightarrow w \in L_2. \end{aligned}$$

(II) Let us set:

$$L_1 = (x,y)^{-1}(\overleftarrow{\mathrm{H}_k}(L)),$$
$$L_2 = \begin{cases} \overleftarrow{\mathrm{H}_k}(x^{-1}(L)) \cup \mathrm{H}'_{k-1}((x,y)^{-1}(L)) & \text{if } k > 1, \\ \overleftarrow{\mathrm{H}_k}(x^{-1}(L)) \cup (x,y)^{-1}(L) & \text{otherwise.} \end{cases}$$

Then

$$\begin{aligned} w \in L_1 &\Leftrightarrow xwy \in \overleftarrow{\mathrm{H}_k}(L) \\ &\Leftrightarrow \begin{cases} (xwy = x\alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha)y \wedge y = \mathrm{H}(x) \wedge \beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha)y \in L \wedge |\beta| = k) \\ \vee (xwy = x\beta\gamma\mathrm{H}(\beta)y \wedge y = \mathrm{H}(x) \wedge x\beta\gamma\mathrm{H}(\beta)y \in L \wedge |\beta| = k-1) \end{cases} \\ &\Leftrightarrow \begin{cases} (w = \alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \wedge y = \mathrm{H}(x) \wedge \beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \in (L)y^{-1} \wedge |\beta| = k) \\ \vee (w = \beta\gamma\mathrm{H}(\beta) \wedge y = \mathrm{H}(x) \wedge \beta\gamma\mathrm{H}(\beta) \in (x,y)^{-1}(L) \wedge |\beta| = k-1) \end{cases} \\ &\Leftrightarrow \begin{cases} (w = \alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \wedge y = \mathrm{H}(x) \wedge w \in \overleftarrow{\mathrm{H}_k}((L)y^{-1})) \\ \vee (w = \beta\gamma\mathrm{H}(\beta) \wedge y = \mathrm{H}(x) \wedge w \in \mathrm{H}'_{k-1}((x,y)^{-1}(L)) \wedge k \neq 1) \\ \vee (w = \gamma \wedge y = \mathrm{H}(x) \wedge w \in (x,y)^{-1}(L) \wedge k = 1) \end{cases} \\ &\Leftrightarrow w \in L_2. \end{aligned}$$

(III) Let us set:

$$L_1 = (x,y)^{-1}(\mathrm{H}'_k(L)),$$
$$L_2 = \mathrm{H}'_{k-1}((x,y)^{-1}(L)),$$
$$L_3 = (x,y)^{-1}(L).$$

Then:

$$
\begin{aligned}
w \in L_1 &\Leftrightarrow xwy \in \mathrm{H}'_k(L) \\
&\Leftrightarrow \left\{ \begin{array}{l} xwy = x\beta\gamma\mathrm{H}(\beta)y \\ \wedge y = \mathrm{H}(x) \\ \wedge x\beta\gamma\mathrm{H}(\beta)y \in L \\ \wedge |\beta| = k-1 \end{array} \right. \\
&\Leftrightarrow \left\{ \begin{array}{l} w = \beta\gamma\mathrm{H}(\beta) \\ \wedge y = \mathrm{H}(x) \\ \wedge \beta\gamma\mathrm{H}(\beta) \in (x,y)^{-1}(L) \\ \wedge |\beta| = k-1 \end{array} \right. \\
&\Leftrightarrow \left\{ \begin{array}{l} (w = \beta\gamma\mathrm{H}(\beta) \wedge y = \mathrm{H}(x) \wedge w \in \mathrm{H}'_{k-1}((x,y)^{-1}(L)) \wedge k > 1) \\ \vee (w \in (x,y)^{-1}(L) \wedge k = 1) \end{array} \right. \\
&\Leftrightarrow \left\{ \begin{array}{l} (w \in L_2 \wedge k > 1) \\ \vee (w \in L_3 \wedge k = 1) \end{array} \right.
\end{aligned}
$$

□

The problem of two-sided residuals of an hairpin completion w.r.t. couples (x, y) with either x or y equal to ε is that they add one catenation that has to be memorized. It can be checked that this may lead to infinite sets of two-sided residuals.

Proposition 5. *Let Γ be an alphabet and H be an anti-morphism over Γ^*. Let L be a language over an alphabet Γ. Let $k > 0$ be an integer. Let L' be a language in $\{\overleftarrow{\mathrm{H}_k}(L), \overrightarrow{\mathrm{H}_k}(L), \mathrm{H}'_k(L)\}$. Let x be a symbol in Γ. Then:*

$$(x,\varepsilon)^{-1}(L') = (x,\mathrm{H}(x))^{-1}(L') \cdot \{\mathrm{H}(x)\},$$
$$(\varepsilon,x)^{-1}(L') = \bigcup_{z\in\Gamma|\mathrm{H}(z)=x} \{z\} \cdot (z,x)^{-1}(L').$$

Proof. Directly deduced from Lemma 1 and from Corollary 3. □

Let L be a language over an alphabet Γ. The set $\mathcal{R}_L$ *of two-sided residuals* of L is defined by: $\mathcal{R}_L = \bigcup_{k\geq 1} \mathcal{R}_L^k$, where

$$\mathcal{R}_L^k = \left\{ \begin{array}{ll} \{(x,y)^{-1}(L) \mid (x,y) \in \Sigma_\Gamma\} & \text{if } k = 1, \\ \{(x,y)^{-1}(L') \mid (x,y) \in \Sigma_\Gamma \wedge\ L' \in \mathcal{R}_L^{k-1}\} & \text{otherwise.} \end{array} \right.$$

From now on we focus on hairpin completion of regular languages. Let us recall that such a completion is not necessarily regular [9].

Lemma 6. *The family of regular languages is not closed under hairpin completion.*

Proof. Let $\Gamma = \{a, b, c\}$, $k > 0$ be a fixed integer and H be the anti-morphism over Γ^* defined by $\mathrm{H}(a) = a$, $\mathrm{H}(b) = c$ and $\mathrm{H}(c) = b$. Let $L' = \overrightarrow{\mathrm{H}_k}(L(a^*b^kc^k))$. Let us first show that $L' = \{a^nb^kc^ka^n \mid n \geq 0\}$. Let w be a word in Γ^*.

$$
\begin{aligned}
w \in L' &\Leftrightarrow w = \alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \wedge \alpha\beta\gamma\mathrm{H}(\beta) \in L(a^*b^kc^k) \wedge |\beta| = k \\
&\Leftrightarrow w = \alpha\beta\gamma\mathrm{H}(\beta)\mathrm{H}(\alpha) \wedge \alpha \in L(a^*) \wedge \mathrm{H}(\beta) = c^k \wedge \beta = b^k \\
&\Leftrightarrow w = a^nb^kc^ka^n \text{ with } n \geq 0.
\end{aligned}
$$

For any integer $j \geq 0$, let us define the language L'_j by:

$$L'_j = \left\{ \begin{array}{ll} L' & \text{if } j = 0, \\ a^{-1}(L'_{j-1}) & \text{otherwise.} \end{array} \right.$$

Consequently, it holds $L'_j = \{a^{n-j}b^kc^ka^n \mid n \geq j\}$. Finally, since for any two distinct integers j and j', the word $b^kc^ka^j$ belongs to $L'_j \setminus L'_{j'}$, it holds that for any two distinct integers j and j', $L'_j \neq L'_{j'}$ and $(a^j)^{-1}(L') \neq (a^{j'})^{-1}(L')$. As a consequence, the set of left residuals of L' is infinite. □

The set of two-sided residuals of a hairpin completion of a regular language may be infinite, but the restriction to residuals w.r.t. couples (x, y) of symbols is sufficient to obtain a finite set of two-sided residuals and a finite recognizer.

5 The Two-Sided Derived Term Automaton

The computation of residuals is intractable when it is defined over languages. However, derived terms of regular expressions denote residuals of regular languages. We then extend the partial derivation of regular expressions [1] to the partial derivation of hairpin expressions.

A *hairpin expression* E over an alphabet Γ is a regular expression over Γ or is inductively defined by: $E = \overleftarrow{\mathrm{H}_k}(F)$, $E = \overrightarrow{\mathrm{H}_k}(F)$, $E = \mathrm{H}'_k(F)$, $E = G_1 + G_2$, where H is any anti-morphism over Γ^*, $k > 0$ is any integer, F is any regular expression over Γ, and G_1 and G_2 are any two hairpin expressions over Σ. If the only operators appearing in E are regular operators ($+$, $\cdot$ or *), the expression E is said to be a *simple hairpin expression*. The *language* denoted by a hairpin expression E over an alphabet Γ is the regular language $L(E)$ if E is a regular expression or is inductively defined by: $L(\overleftarrow{\mathrm{H}_k}(F)) = \overleftarrow{\mathrm{H}_k}(L(F))$, $L(\overrightarrow{\mathrm{H}_k}(F)) = \overrightarrow{\mathrm{H}_k}(L(F))$, $L(\mathrm{H}'_k(F)) = \mathrm{H}'_k(L(F))$, $L(G_1 + G_2) = L(G_1) \cup L(G_2)$, where H is any anti-morphism over Γ^*, $k > 0$ is any integer, F is any regular expression over Γ, and G_1 and G_2 are any two hairpin expressions over Γ.

Definition 6. *Let E be a hairpin expression over an alphabet Γ. Let (x, y) be a couple of symbols in Σ_Γ. Let $k > 0$ be an integer. The* two-sided partial derivative *of E* w.r.t. *(x, y) is the set $\frac{\partial}{\partial_{(x,y)}}(E)$ of hairpin expressions defined by:*

$$\frac{\partial}{\partial_{(x,y)}}(F) = \begin{cases} (F)\frac{\partial}{\partial_y} & \text{if } x = \varepsilon, \\ \frac{\partial}{\partial_x}(F) & \text{if } y = \varepsilon, \\ \bigcup_{F' \in \frac{\partial}{\partial_x}(F)} (F')\frac{\partial}{\partial_y} & \text{otherwise,} \end{cases}$$

$$\frac{\partial}{\partial_{(x,y)}}(\overrightarrow{\mathrm{H}_k}(F)) = \begin{cases} \emptyset & \text{if } y \neq \mathrm{H}(x), \\ \overrightarrow{\mathrm{H}_k}(\frac{\partial}{\partial_x}(F)) \cup \frac{\partial}{\partial_{(x,y)}}(F) & \text{if } y = \mathrm{H}(x) \ \wedge \ k = 1 \\ \overrightarrow{\mathrm{H}_k}(\frac{\partial}{\partial_x}(F)) \cup \mathrm{H}'_{k-1}(\frac{\partial}{\partial_{(x,y)}}(F)) & \text{otherwise,} \end{cases}$$

$$\frac{\partial}{\partial_{(x,y)}}(\overleftarrow{\mathrm{H}_k}(F)) = \begin{cases} \emptyset & \text{if } y \neq \mathrm{H}(x), \\ \overleftarrow{\mathrm{H}_k}((F)\frac{\partial}{\partial_y}) \cup \frac{\partial}{\partial_{(x,y)}}(F) & \text{if } y = \mathrm{H}(x) \ \wedge \ k = 1 \\ \overleftarrow{\mathrm{H}_k}((F)\frac{\partial}{\partial_y}) \cup \mathrm{H}'_{k-1}(\frac{\partial}{\partial_{(x,y)}}(F)) & \text{otherwise,} \end{cases}$$

$$\frac{\partial}{\partial_{(x,y)}}(\mathrm{H}'_k(F)) = \begin{cases} \emptyset & \text{if } y \neq \mathrm{H}(x), \\ \mathrm{H}'_{k-1}(\frac{\partial}{\partial_{(x,y)}}(F)) & \text{if } k{>}1, \\ \frac{\partial}{\partial_{(x,y)}}(F) & \text{otherwise,} \end{cases}$$

$$\frac{\partial}{\partial_{(x,y)}}(G_1 + G_2) = \frac{\partial}{\partial_{(x,y)}}(G_1) \cup \frac{\partial}{\partial_{(x,y)}}(G_2),$$

where H *is any anti-morphism over* Γ^*, $k > 0$ *is any integer,* F *is any regular expression over* Γ, G_1 *and* G_2 *are any two hairpin expressions over* Γ, *and for any set* $\mathcal{H}$ *of hairpin expressions:* $\overrightarrow{\mathrm{H}_k}(\mathcal{H}) = \{\overrightarrow{\mathrm{H}_k}(H) \mid H \in \mathcal{H}\}$, $\overleftarrow{\mathrm{H}_k}(\mathcal{H}) = \{\overleftarrow{\mathrm{H}_k}(H) \mid H \in \mathcal{H}\}$, $\mathrm{H}'_k(\mathcal{H}) = \{\mathrm{H}'_k(H) \mid H \in \mathcal{H}\}$.

Let E be a hairpin expression over an alphabet Γ. The *set* $\overleftrightarrow{\mathcal{D}_E}$ *of two-sided derived terms of the expression* E is defined by: $\overleftrightarrow{\mathcal{D}_E} = \bigcup_{k\geq 1} \overleftrightarrow{\mathcal{D}^k_E}$, where:

$$\overleftrightarrow{\mathcal{D}^k_E} = \begin{cases} \bigcup_{(x,y)\in\Sigma_\Gamma} \frac{\partial}{\partial(x,y)}(E) & \text{if } k = 1, \\ \bigcup_{(x,y)\in\Sigma_\Gamma, E'\in\overleftrightarrow{\mathcal{D}^{k-1}_E}} \frac{\partial}{\partial(x,y)}(E') & \text{otherwise.} \end{cases}$$

Derived terms of regular expressions are related to left residuals. Let us show that derived terms of hairpin expressions are related to two-sided residuals.

Proposition 6. *Let* E *be a hairpin expression over an alphabet* Γ. *Let* (x, y) *be a couple of symbols in* Γ^2. *Then:* $\bigcup_{F\in\frac{\partial}{\partial_{(x,y)}}(E)} L(F) = (x, y)^{-1}(L(E))$.

Furthermore, if E *is a regular expression, the proposition still holds whenever* (x, y) *is a couple of symbols in* Σ_Γ.

Proof. Trivially proved by induction over the structure of E, according to Proposition 4. □

Determining whether the empty word belongs to the language denoted by a regular expression E can be performed syntactically and inductively as follows:

$$\varepsilon \notin L(a),\ \varepsilon \notin L(\emptyset),\ \varepsilon \in L(\varepsilon),$$
$$\varepsilon \in L(G_1 \cdot G_2) \Leftrightarrow \varepsilon \in L(G_1) \ \wedge\ \varepsilon \in L(G_2),$$
$$\varepsilon \in L(G_1 + G_2) \Leftrightarrow \varepsilon \in L(G_1) \ \vee\ \varepsilon \in L(G_2),\ \varepsilon \in L(G_1^*).$$

This syntactical test is needed to compute the derived term automaton since it defines the finality of the states. We now show how to extend this computation to hairpin expressions.

Lemma 7. *Let* F *be a regular expression and* G_1 *and* G_2 *be two hairpin expressions. Then:*

$$\varepsilon \notin L(\overrightarrow{\mathrm{H}_k}(F)),\ \varepsilon \notin L(\overleftarrow{\mathrm{H}_k}(F)),\ \varepsilon \notin L(\mathrm{H}'_k(F)),$$
$$\varepsilon \in L(G_1 + G_2) \Leftrightarrow \varepsilon \in L(G_1) \ \vee\ \varepsilon \in L(G_2).$$

Proof. Trivially proved according to Definition 4, Definition 5 and definition of languages denoted by hairpin expressions. □

The following example illustrates the computation of derived terms. For clarity, in this example, we assume that hairpin expressions are quotiented w.r.t. the following rules: $\varepsilon \cdot E \sim E$, $\emptyset \cdot E \sim \emptyset$. Moreover, sets of expressions are also quotiented w.r.t. the following rule: $\{\emptyset\} \sim \emptyset$.

Example 2. *Let* $\Gamma = \{a, b, c\}$ *and* H *be the anti-morphism over* Γ^* *defined by* $\mathrm{H}(a) = a$, $\mathrm{H}(b) = c$ *and* $\mathrm{H}(c) = b$. *Let* $E = \overrightarrow{\mathrm{H}_1}(a^*bc)$. *Derived terms of* E *are computed as follows:*

$$\frac{\partial}{\partial(a,a)}(E) = \{E\},$$
$$\frac{\partial}{\partial(b,c)}(E) = \{\overrightarrow{\mathrm{H}_1}(c), \varepsilon\},$$
$$\frac{\partial}{\partial(c,b)}(\overrightarrow{\mathrm{H}_1}(c)) = \{\overrightarrow{\mathrm{H}_1}(\varepsilon)\}.$$

Other partial derivatives are equal to $\emptyset$. *Furthermore, it holds that* ε *is the only derived term* F *of* E *such that* ε *belongs to* $L(F)$.

In the following we are looking for an upper bound over the cardinality of the set of two-sided derived terms, thus we apply no reduction to the regular expressions. Notice that this cardinality decreases whenever any reduction is applied.

Lemma 8. *Let* E *and* F *be two regular expressions over an alphabet* Γ. *Then the three following propositions hold:*

1. $\overleftrightarrow{\mathcal{D}_{E+F}} \subset \overleftrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_F}$,
2. $\overleftrightarrow{\mathcal{D}_{E\cdot F}} \subset \overleftarrow{\mathcal{D}_E} \cdot \overrightarrow{\mathcal{D}_F} \cup \overleftrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_F}$,
3. $\overleftrightarrow{\mathcal{D}_{E^*}} \subset \overleftarrow{\mathcal{D}_E} \cdot E^* \cup E^* \cdot \overrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_E} \cup (\overleftarrow{\mathcal{D}_E} \cdot E^*) \cdot \overrightarrow{\mathcal{D}_E} \cup \overleftarrow{\mathcal{D}_E} \cdot (E^* \cdot \overrightarrow{\mathcal{D}_E})$.

Furthermore, $\overleftrightarrow{\mathcal{D}_\varepsilon} = \overleftrightarrow{\mathcal{D}_\emptyset} = \emptyset$ *and* $\overleftrightarrow{\mathcal{D}_a} = \{\varepsilon\}$ *for any symbol* a *in* Γ.

Proof. Basic cases (ε, $\emptyset$ and a in Γ) are trivially proved directly applying Definition 6.

By induction over the structure of the set of two-sided derived terms. Suppose that E and F are two regular expressions over an alphabet Γ. Let (x, y) be a couple of symbols in Σ_Γ.

1. Let us first show that $\frac{\partial}{\partial(x,y)}(E+F) \subset \overleftrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_F}$. According to Definition 6, it holds:
$$\frac{\partial}{\partial(x,y)}(E+F) = \begin{cases} \frac{\partial}{\partial_x}(E+F) & \text{if } y = \varepsilon, \\ (E+F)\frac{\partial}{\partial_y} & \text{if } x = \varepsilon, \\ \bigcup_{G \in \frac{\partial}{\partial_x}(E+F)}(G)\frac{\partial}{\partial_y} & \text{otherwise.} \end{cases}$$
$$= \begin{cases} \frac{\partial}{\partial_x}(E) \cup \frac{\partial}{\partial_x}(F) & \text{if } y = \varepsilon, \\ (E)\frac{\partial}{\partial_y} \cup (F)\frac{\partial}{\partial_y} & \text{if } x = \varepsilon, \\ \bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G)\frac{\partial}{\partial_y} \cup \bigcup_{G \in \frac{\partial}{\partial_x}(F)}(G)\frac{\partial}{\partial_y} & \text{otherwise.} \end{cases}$$
Notice that the three following conditions hold:
$$\frac{\partial}{\partial_x}(E) \cup \frac{\partial}{\partial_x}(F) \subset \overleftarrow{\mathcal{D}_E} \cup \overleftarrow{\mathcal{D}_F} \subset \overleftrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_F},$$
$$(E)\frac{\partial}{\partial_y} \cup (F)\frac{\partial}{\partial_y} \subset \overrightarrow{\mathcal{D}_E} \cup \overrightarrow{\mathcal{D}_F} \subset \overleftrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_F},$$
$$\bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G)\frac{\partial}{\partial_y} \cup \bigcup_{G \in \frac{\partial}{\partial_x}(F)}(G)\frac{\partial}{\partial_y} = \frac{\partial}{\partial(x,y)}(E) \cup \frac{\partial}{\partial(x,y)}(F) \subset \overleftrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_F}.$$
As a consequence, $\frac{\partial}{\partial(x,y)}(E+F) \subset \overleftrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_F}$.

Furthermore, since by definition of the sets of two-sided derived terms, for any expression G in $\overleftrightarrow{\mathcal{D}_E}$ (resp. in $\overleftrightarrow{\mathcal{D}_F}$), $\frac{\partial}{\partial(x,y)}(G) \subset \overleftrightarrow{\mathcal{D}_E}$ (resp. $\frac{\partial}{\partial(x,y)}(G) \subset \overleftrightarrow{\mathcal{D}_F}$), the proposition is satisfied.

2. Let us set $\mathcal{E} = \overleftarrow{\mathcal{D}_E} \cdot \overrightarrow{\mathcal{D}_F} \cup \overleftrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_F}$.

   (a) Let us first show that $\frac{\partial}{\partial(x,y)}(E \cdot F) \subset \frac{\partial}{\partial_x}(E) \cdot (F)\frac{\partial}{\partial_y} \cup \frac{\partial}{\partial(x,y)}(E) \cup \frac{\partial}{\partial(x,y)}(F) \subset \mathcal{E}$.
   According to Definition 6, it holds:

$$\frac{\partial}{\partial(x,y)}(E \cdot F) = \begin{cases} \frac{\partial}{\partial_x}(E \cdot F) & \text{if } y = \varepsilon, \\ (E \cdot F)\frac{\partial}{\partial_y} & \text{if } x = \varepsilon, \\ \bigcup_{G \in \frac{\partial}{\partial_x}(E \cdot F)}(G)\frac{\partial}{\partial_y} & \text{otherwise.} \end{cases}$$

$$= \begin{cases} \frac{\partial}{\partial_x}(E) \cdot F \cup \frac{\partial}{\partial_x}(F) & \text{if } y = \varepsilon \ \wedge \varepsilon \in L(E), \\ \frac{\partial}{\partial_x}(E) \cdot F & \text{if } y = \varepsilon \ \wedge \varepsilon \notin L(E), \\ (E)\frac{\partial}{\partial_y} \cup E \cdot (F)\frac{\partial}{\partial_y} & \text{if } x = \varepsilon \ \wedge \varepsilon \in L(F), \\ E \cdot (F)\frac{\partial}{\partial_y} & \text{if } x = \varepsilon \ \wedge \varepsilon \notin L(F), \\ \bigcup_{G \in \frac{\partial}{\partial_x}(E) \cdot F}(G)\frac{\partial}{\partial_y} \cup \bigcup_{G \in \frac{\partial}{\partial_x}(F)}(G)\frac{\partial}{\partial_y} & \text{if } x, y \in \Gamma \ \wedge \ \varepsilon \in L(E) \\ \bigcup_{G \in \frac{\partial}{\partial_x}(E) \cdot F}(G)\frac{\partial}{\partial_y} & \text{otherwise.} \end{cases}$$

   Notice that the three following conditions hold:

$$\frac{\partial}{\partial_x}(E) \cdot F \cup \frac{\partial}{\partial_x}(F) \subset \overleftarrow{\mathcal{D}_E} \cdot \overrightarrow{\mathcal{D}_F} \cup \overleftrightarrow{\mathcal{D}_F},$$
$$(E)\frac{\partial}{\partial_y} \cup E \cdot (F)\frac{\partial}{\partial_y} \subset \overleftrightarrow{\mathcal{D}_E} \cup \overleftarrow{\mathcal{D}_E} \cdot \overrightarrow{\mathcal{D}_F},$$
$$\bigcup_{G \in \frac{\partial}{\partial_x}(F)}(G)\frac{\partial}{\partial_y} = \frac{\partial}{\partial(x,y)}(F) \subset \overrightarrow{\mathcal{D}_F}.$$

   Moreover,

$$\bigcup_{G \in \frac{\partial}{\partial_x}(E) \cdot F}(G)\frac{\partial}{\partial_y} = \begin{cases} \bigcup_{G \in \frac{\partial}{\partial_x}(E)} G \cdot (F)\frac{\partial}{\partial_y} \cup \bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G)\frac{\partial}{\partial_y} & \text{if } \varepsilon \in L(F), \\ \bigcup_{G \in \frac{\partial}{\partial_x}(E)} G \cdot (F)\frac{\partial}{\partial_y} & \text{otherwise.} \end{cases}$$

   Finally, since $\bigcup_{G \in \frac{\partial}{\partial_x}(E)} G \cdot (F)\frac{\partial}{\partial_y} = \frac{\partial}{\partial_x}(E) \cdot (F)\frac{\partial}{\partial_y} \subset \overleftarrow{\mathcal{D}_E} \cdot \overrightarrow{\mathcal{D}_F}$ and since $\bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G)\frac{\partial}{\partial_y} = \frac{\partial}{\partial(x,y)}(E) \subset \overrightarrow{\mathcal{D}_E}$, the proposition is satisfied.

   (b) Let us now show that for any expression G in $\mathcal{E}$, $\frac{\partial}{\partial(x,y)}(G) \subset \mathcal{E}$.

      i. if G belongs to $\overleftrightarrow{\mathcal{D}_E}$ (resp. to $\overleftrightarrow{\mathcal{D}_F}$), by definition of the set of two-sided derived terms it holds $\frac{\partial}{\partial(x,y)}(G) \subset \overleftrightarrow{\mathcal{D}_E}$ (resp. $\frac{\partial}{\partial(x,y)}(G) \subset \overleftrightarrow{\mathcal{D}_F}$).

      ii. If G belongs to $\overleftarrow{\mathcal{D}_E} \cdot \overrightarrow{\mathcal{D}_F}$, then $G = G_1 \cdot G_2$ and from **(2a)** it holds that $\frac{\partial}{\partial(x,y)}(G) \subset \overleftarrow{\mathcal{D}_{G_1}} \cdot \overrightarrow{\mathcal{D}_{G_2}} \cup \overleftrightarrow{\mathcal{D}_{G_1}} \cup \overleftrightarrow{\mathcal{D}_{G_2}}$. According to definition of the set of two-sided derived terms, the four follwong conditions hold:
      $\overleftarrow{\mathcal{D}_{G_1}} \subset \overleftarrow{\mathcal{D}_E}$, $\overleftrightarrow{\mathcal{D}_{G_1}} \subset \overleftrightarrow{\mathcal{D}_E}$, $\overrightarrow{\mathcal{D}_{G_2}} \subset \overrightarrow{\mathcal{D}_F}$ and $\overleftrightarrow{\mathcal{D}_{G_2}} \subset \overleftrightarrow{\mathcal{D}_F}$.

   As a consequence, the proposition is satisfied.

3. Let us set $\mathcal{E} = \overleftarrow{\mathcal{D}_E} \cdot E^* \cup E^* \cdot \overrightarrow{\mathcal{D}_E} \cup \overleftrightarrow{\mathcal{D}_E} \cup (\overleftarrow{\mathcal{D}_E} \cdot E^*) \cdot \overrightarrow{\mathcal{D}_E} \cup \overleftarrow{\mathcal{D}_E} \cdot (E^* \cdot \overrightarrow{\mathcal{D}_E})$.

(a) Let us first show that $\frac{\partial}{\partial(x,y)}(E^*) \subset \mathcal{E}$. According to Definition 6, it holds:

$$\frac{\partial}{\partial(x,y)}(E^*) = \begin{cases} \frac{\partial}{\partial_x}(E^*) & \text{if } y = \varepsilon, \\ (E^*)\frac{\partial}{\partial_y} & \text{if } x = \varepsilon, \\ \bigcup_{G \in \frac{\partial}{\partial_x}(E^*)}(G)\frac{\partial}{\partial_y} & \text{otherwise.} \end{cases}$$
$$= \begin{cases} \frac{\partial}{\partial_x}(E) \cdot E^* & \text{if } y = \varepsilon, \\ E^* \cdot (E)\frac{\partial}{\partial_y} & \text{if } x = \varepsilon, \\ \bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G \cdot E^*)\frac{\partial}{\partial_y} & \text{otherwise.} \end{cases}$$

Notice that $\frac{\partial}{\partial_x}(E) \cdot E^* \subset \overleftarrow{\mathcal{D}_E} \cdot E^*$ and that $E^* \cdot (E)\frac{\partial}{\partial_y} \subset E^* \cdot \overrightarrow{\mathcal{D}_E}$. Moreover,

$$\bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G \cdot E^*)\frac{\partial}{\partial_y}$$
$$= \bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G)\frac{\partial}{\partial_y} \cup G \cdot (E^*)\frac{\partial}{\partial_y}$$
$$= \bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G)\frac{\partial}{\partial_y} \cup G \cdot (E^* \cdot (E)\frac{\partial}{\partial_y})$$
$$= \bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G)\frac{\partial}{\partial_y} \cup \bigcup_{G \in \frac{\partial}{\partial_x}(E)} G \cdot (E^* \cdot (E)\frac{\partial}{\partial_y})$$

Finally, since the two following conditions hold:

$$\bigcup_{G \in \frac{\partial}{\partial_x}(E)}(G)\frac{\partial}{\partial_y} = \frac{\partial}{\partial(x,y)}(E) \subset \overleftrightarrow{\mathcal{D}_E}$$

and $\bigcup_{G \in \frac{\partial}{\partial_x}(E)} G \cdot (E^* \cdot (E)\frac{\partial}{\partial_y}) = \frac{\partial}{\partial_x}(E) \cdot (E^* \cdot (E)\frac{\partial}{\partial_y}) \subset \overleftarrow{\mathcal{D}_E} \cdot (E^* \cdot \overrightarrow{\mathcal{D}_E})$, it holds that $\frac{\partial}{\partial(x,y)}(E^*) \subset \mathcal{E}$.

(b) Let us now show that for any expression G in $\mathcal{E}$, $\frac{\partial}{\partial(x,y)}(G) \subset \mathcal{E}$.

i. if G belongs to $\overleftrightarrow{\mathcal{D}_E}$, by definition of the set of two-sided derived terms it holds $\frac{\partial}{\partial(x,y)} \subset \overleftrightarrow{\mathcal{D}_E}$.

ii. if G belongs to $\overleftarrow{\mathcal{D}_E} \cdot E^*$, then $G = G_1 \cdot E^*$ and from **(2a)** it holds that:
$\frac{\partial}{\partial(x,y)}(G) \subset \frac{\partial}{\partial_x}(G_1) \cdot (E^*)\frac{\partial}{\partial_y} \cup \frac{\partial}{\partial(x,y)}(G_1) \cup \frac{\partial}{\partial(x,y)}(E^*)$.
Moreover, since from **(3a)** $\frac{\partial}{\partial(x,y)}(E^*) \subset \mathcal{E}$, since $\frac{\partial}{\partial_x}(G_1) \subset \overleftarrow{\mathcal{D}_E}$ and since $\frac{\partial}{\partial(x,y)}(G_1) \subset \overleftrightarrow{\mathcal{D}_E}$, it holds that:

$$\frac{\partial}{\partial(x,y)}(G)$$
$$\subset \overleftarrow{\mathcal{D}_E} \cdot (E^* \cdot (E)\frac{\partial}{\partial_y}) \cup \overleftrightarrow{\mathcal{D}_E} \cup \mathcal{E}$$
$$\subset \overleftarrow{\mathcal{D}_E} \cdot (E^* \cdot \overrightarrow{\mathcal{D}_E}) \cup \overleftrightarrow{\mathcal{D}_E} \cup \mathcal{E}$$
$$\subset \mathcal{E}$$

iii. if G belongs to $E^* \cdot \overrightarrow{\mathcal{D}_E}$, then $G = E^* \cdot G_1$ and from **(2a)** it holds that:
$\frac{\partial}{\partial(x,y)}(G) \subset \frac{\partial}{\partial_x}(E^*) \cdot (G_1)\frac{\partial}{\partial_y} \cup \frac{\partial}{\partial(x,y)}(E^*) \cup \frac{\partial}{\partial(x,y)}(G_1)$.
Moreover, since from **(3a)** $\frac{\partial}{\partial(x,y)}(E^*) \subset \mathcal{E}$, since $(G_1)\frac{\partial}{\partial_y} \subset \overrightarrow{\mathcal{D}_E}$ and since $\frac{\partial}{\partial(x,y)}(G_1) \subset \overleftrightarrow{\mathcal{D}_E}$:

$$\frac{\partial}{\partial(x,y)}(G)$$

$$\subset (\tfrac{\partial}{\partial_x}(E)\cdot E^*)\cdot\overrightarrow{\mathcal{D}_E}\cup\mathcal{E}\cup\overleftrightarrow{\mathcal{D}_E}$$
$$\subset (\overleftarrow{\mathcal{D}_E}\cdot E^*)\cdot\overrightarrow{\mathcal{D}_E}\cup\mathcal{E}\cup\overleftrightarrow{\mathcal{D}_E}$$
$$\subset \mathcal{E}$$

iv. If G belongs to $(\overleftarrow{\mathcal{D}_E}\cdot E^*)\cdot\overrightarrow{\mathcal{D}_E}$, then $G=(G_1\cdot E^*)\cdot G_2$ and from **(2a)** it holds that:

$$\tfrac{\partial}{\partial(x,y)}(G)\subset\tfrac{\partial}{\partial_x}(G_1\cdot E^*)\cdot(G_2)\tfrac{\partial}{\partial_y}\cup\tfrac{\partial}{\partial(x,y)}(G_1\cdot E^*)\cup\tfrac{\partial}{\partial(x,y)}(G_2).$$

Since $\frac{\partial}{\partial_x}(G_1\cdot E^*)\subset\frac{\partial}{\partial_x}(G_1)\cdot E^*\cup\frac{\partial}{\partial_x}(E^*)$, it holds that:

$$\tfrac{\partial}{\partial_x}(G_1\cdot E^*)\cdot(G_2)\tfrac{\partial}{\partial_y}\subset(\tfrac{\partial}{\partial_x}(G_1)\cdot E^*)\cdot(G_2)\tfrac{\partial}{\partial_y}\cup(\tfrac{\partial}{\partial_x}(E)\cdot E^*)\cdot(G_2)\tfrac{\partial}{\partial_y}.$$

Finally, since from **(3bii)** $\frac{\partial}{\partial(x,y)}(G_1\cdot E^*)\subset\mathcal{E}$, it holds:

$$\tfrac{\partial}{\partial(x,y)}(G)$$
$$\subset(\overleftarrow{\mathcal{D}_E}\cdot E^*)\cdot\overrightarrow{\mathcal{D}_E}\cup\mathcal{E}\cup\overleftrightarrow{\mathcal{D}_E}$$
$$\subset\mathcal{E}$$

v. If G belongs to $\overleftarrow{\mathcal{D}_E}\cdot(E^*\cdot\overrightarrow{\mathcal{D}_E})$, then $G=G_1\cdot(E^*\cdot G_2)$ and from **(2a)** it holds that:

$$\tfrac{\partial}{\partial(x,y)}(G)\subset\tfrac{\partial}{\partial_x}(G_1)\cdot(E^*\cdot G_2)\tfrac{\partial}{\partial_y}\cup\tfrac{\partial}{\partial(x,y)}(G_1)\cup\tfrac{\partial}{\partial(x,y)}(E^*\cdot G_2).$$

Since $(E^*\cdot G_2)\frac{\partial}{\partial_y}\subset(E^*)\frac{\partial}{\partial_y}\cup E^*\cdot(G_2)\frac{\partial}{\partial_y}$, it holds that:

$$\tfrac{\partial}{\partial_x}(G_1)\cdot(E^*\cdot G_2)\tfrac{\partial}{\partial_y}\subset\tfrac{\partial}{\partial_x}(G_1)\cdot(E^*)\tfrac{\partial}{\partial_y}\cup\tfrac{\partial}{\partial_x}(G_1)\cdot(E^*\cdot(G_2)\tfrac{\partial}{\partial_y}).$$

Finally, since from **(3biii)** $\frac{\partial}{\partial(x,y)}(E^*\cdot G_2)\subset\mathcal{E}$, it holds:

$$\tfrac{\partial}{\partial(x,y)}(G)$$
$$\subset\overleftarrow{\mathcal{D}_E}\cdot(E^*\cdot\overrightarrow{\mathcal{D}_E})\cup\mathcal{E}\cup\overleftrightarrow{\mathcal{D}_E}$$
$$\subset\mathcal{E}$$

As a consequence, the proposition is satisfied.

□

Proposition 7. *Let E be a regular expression of width $n>0$ and of star number h. Let us set $m=n+h$. Then the three following propositions hold:*

1. $\mathrm{Card}(\overleftarrow{\mathcal{D}_E})\leq n$,
2. $\mathrm{Card}(\overrightarrow{\mathcal{D}_E})\leq n$,
3. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_E})\leq\frac{2m\times(m+1)\times(m+2)}{3}-3$.

Proof. For the set of left derived terms, the proposition is proved in [1], where it is shown that the cardinality of the set $\{E'\mid\exists w\in\Sigma^+,E'\in\frac{\partial}{\partial_w}(E)\}$ is less than n. This bound still holds for the set of right derived terms.

Let n_1 (resp. n_2) be the width of a regular expression F (resp. G) and h_1 (resp. h_2) be the star number of F (resp. G). Let us set $m_1=n_1+h_1$ and $m_2=n_2+h_2$. For $E=F+G$ and for $E=F\cdot G$, we have $n=n_1+n_2$, $h=h_1+h_2$ and $m=m_1+m_2$. For $E=F^*$, we have $n=n_1$, $h=h_1+1$ and $m=m_1+1$.

According to Lemma 8, we get:

1. $\overleftrightarrow{\mathcal{D}_{F+G}} \subset \overleftrightarrow{\mathcal{D}_F} \cup \overleftrightarrow{\mathcal{D}_G}$,
2. $\overleftrightarrow{\mathcal{D}_{F\cdot G}} \subset \overleftarrow{\mathcal{D}_F} \cdot \overrightarrow{\mathcal{D}_G} \cup \overleftrightarrow{\mathcal{D}_F} \cup \overleftrightarrow{\mathcal{D}_G}$,
3. $\overleftrightarrow{\mathcal{D}_{F^*}} \subset \overleftarrow{\mathcal{D}_F} \cdot F^* \cup F^* \cdot \overrightarrow{\mathcal{D}_F} \cup \overleftrightarrow{\mathcal{D}_F} \cup (\overleftarrow{\mathcal{D}_F} \cdot F^*) \cdot \overrightarrow{\mathcal{D}_F} \cup \overleftarrow{\mathcal{D}_F} \cdot (F^* \cdot \overrightarrow{\mathcal{D}_F})$.

As a consequence, we get:

1. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{F+G}}) \leq \mathrm{Card}(\overleftrightarrow{\mathcal{D}_F}) + \mathrm{Card}(\overleftrightarrow{\mathcal{D}_G})$,
2. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{F\cdot G}}) \leq \mathrm{Card}(\overleftrightarrow{\mathcal{D}_F}) + \mathrm{Card}(\overleftrightarrow{\mathcal{D}_G}) + n_1 n_2$,
3. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{F^*}}) \leq \mathrm{Card}(\overleftrightarrow{\mathcal{D}_F}) + 2n_1(n_1+1)$.

On the one hand the cardinality of $\overleftrightarrow{\mathcal{D}_{F^*}}$ is strictly greater than the cardinality of $\overleftrightarrow{\mathcal{D}_F}$ although F and F^* have the same width n_1; we therefore substitute the parameter $m_1 = n_1 + h_1$ to n_1, so that F^* is associated with $m_1 + 1$.

On the other hand, the maximal increase of the cardinality of $\overleftrightarrow{\mathcal{D}_E}$ (w.r.t. m) occurs in the star case; we therefore consider the function ϕ such that:

1. $\phi(0) = 0$ and $\phi(1) = 1$,
2. $\phi(k+1) = \phi(k) + 2 \times k \times (k+1)$,

and we show that $\overleftrightarrow{\mathcal{D}_E} \leq \phi(m)$ for any regular expression E.

According to Lemma 8 and by induction hypothesis, it holds:

1. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{F+G}}) \leq \phi(m_1) + \phi(m_2)$,
2. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{F\cdot G}}) \leq \phi(m_1) + \phi(m_2) + n_1 \times n_2$,
3. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{F^*}}) \leq \phi(m_1) + 2n_1(n_1+1)$.

It can be checked that:

$$\phi(m_1) + \phi(m_2) \leq \phi(m_1) + \phi(m_2) + n_1 \times n_2 \leq \phi(m_1 + m_2).$$

As a consequence, it holds:

1. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{F+G}}) \leq \phi(m_1 + m_2)$,
2. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{F\cdot G}}) \leq \phi(m_1 + m_2)$.

Furthermore, by definition of ϕ and since $m_1 \geq n_1$, it holds:

$$\phi(m_1) + 2n_1(n_1+1) \leq \phi(m_1) + 2(m_1)(m_1+1) = \phi(m_1+1)$$

and consequently $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{F^*}}) \leq \phi(m_1+1)$.

Finally, since $\sum_{j=1}^{k} j(j+1) = \frac{k(k+1)(k+2)}{3}$, it holds for all integer $k \geq 1$:

$$\phi(k) = \tfrac{2k(k+1)(k+2)}{3} - 3.$$

□

Proposition 8. *Let E be a regular expression over an alphabet Γ, H be an antimorphism over Γ^* and $k > 0$ be an integer. Then:*

1. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{\mathrm{H}'_k(E)}}) \leq k \times \mathrm{Card}(\overleftrightarrow{\mathcal{D}_E})$,

2. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{\overrightarrow{\mathrm{H}_k}(E)}}) \leq \mathrm{Card}(\overleftarrow{\mathcal{D}_E}) + k \times \mathrm{Card}(\overleftrightarrow{\mathcal{D}_E})$,

3. $\mathrm{Card}(\overleftrightarrow{\mathcal{D}_{\overleftarrow{\mathrm{H}_k}(E)}}) \leq \mathrm{Card}(\overrightarrow{\mathcal{D}_E}) + k \times \mathrm{Card}(\overleftrightarrow{\mathcal{D}_E})$.

Proof. Let E be a regular expression.

(1) Let us set $\mathcal{E} = \{\mathrm{H}'_{k'}(E') \mid E' \in \overleftrightarrow{\mathcal{D}_E} \ \wedge \ k' < k\} \cup \overleftrightarrow{\mathcal{D}_E}$. Let us show that $\overleftrightarrow{\mathcal{D}_{\mathrm{H}'_k(E)}} \subset \mathcal{E}$.

(a) According to Definition 6, for any couple (x, y) in Σ_Γ, $\frac{\partial}{\partial(x,y)}(\mathrm{H}'_k(E)) \subset \mathcal{E}$.

(b) Let us show that any derived term of an expression G in $\mathcal{E}$ belongs to $\mathcal{E}$.

(i) if G belongs to $\overleftrightarrow{\mathcal{D}_E}$, so do its derived terms.

(ii) if $G \in \{\mathrm{H}'_{k'}(E') \mid E' \in \overleftrightarrow{\mathcal{D}_E} \ \wedge \ k' < k\}$, then $G = \mathrm{H}'_{k'}(G_1)$ with $G_1 \in \overleftrightarrow{\mathcal{D}_E}$ and from Definition 6 it holds:

$$\tfrac{\partial}{\partial(x,y)}(G) \subset \{\mathrm{H}'_{k''}(G_2) \mid G_2 \in \overleftrightarrow{\mathcal{D}_{G_1}} \ \wedge \ k'' < k'\} \cup \overleftrightarrow{\mathcal{D}_{G_1}}.$$

By definition of G_1, $\overleftrightarrow{\mathcal{D}_{G_1}} \subset \overleftrightarrow{\mathcal{D}_E}$. Consequently $\frac{\partial}{\partial(x,y)}(G) \subset \mathcal{E}$.

(c) Finally, since $\mathrm{Card}(\mathcal{E}) = (k-1) \times \mathrm{Card}(\overleftrightarrow{\mathcal{D}_E}) + \mathrm{Card}(\overleftrightarrow{\mathcal{D}_E})$, the proposition holds.

(2) Let us set $\mathcal{E} = \{\overrightarrow{\mathrm{H}_k}(E') \mid E' \in \overleftarrow{\mathcal{D}_E}\} \cup \{\mathrm{H}'_{k'}(E') \mid E' \in \overleftrightarrow{\mathcal{D}_E} \wedge k' < k\} \cup \overleftrightarrow{\mathcal{D}_E}$. Let us show that $\overleftrightarrow{\mathcal{D}_{\overrightarrow{\mathrm{H}_k}(E)}} \subset \mathcal{E}$.

(a) According to Definition 6, for any couple (x, y) in Σ_Γ, $\frac{\partial}{\partial(x,y)}(\overrightarrow{\mathrm{H}_k}(E)) \subset \mathcal{E}$.

(b) Let us show that any derived term of an expression G in $\mathcal{E}$ belongs to $\mathcal{E}$.

(i) if G belongs to $\{\overrightarrow{\mathrm{H}_k}(E') \mid E' \in \overleftarrow{\mathcal{D}_E}\}$ then $G = \overrightarrow{\mathrm{H}_k}(G_1)$ with $G_1 \in \overleftrightarrow{\mathcal{D}_E}$ and from Definition 6 it holds that:

$$\tfrac{\partial}{\partial(x,y)}(G) \subset \{\overrightarrow{\mathrm{H}_k}(G_2) \mid G_2 \in \overleftarrow{\mathcal{D}_{G_1}}\} \cup \{\mathrm{H}'_{k'}(G_2) \mid G_2 \in \overleftrightarrow{\mathcal{D}_{G_1}} \ \wedge \ k' < k\} \cup \overleftrightarrow{\mathcal{D}_{G_1}}.$$

Since by definition of G_1, $\overleftrightarrow{\mathcal{D}_{G_1}} \subset \overleftrightarrow{\mathcal{D}_E}$ and $\overleftarrow{\mathcal{D}_{G_1}} \subset \overleftarrow{\mathcal{D}_E}$, it holds: $\frac{\partial}{\partial(x,y)}(G) \subset \mathcal{E}$.

(ii) if G belongs to $\{\mathrm{H}'_{k'}(E') \mid E' \in \overleftrightarrow{\mathcal{D}_E} \ \wedge \ k' < k\}$,then $G = \mathrm{H}'_{k'}(G_1)$ with $G_1 \in \overleftrightarrow{\mathcal{D}_E}$ and from Definition 6 it holds:

$$\tfrac{\partial}{\partial(x,y)}(G) \subset \{\mathrm{H}'_{k''}(G_2) \mid G_2 \in \overleftrightarrow{\mathcal{D}_{G_1}} \ \wedge \ k'' < k'\} \cup \overleftrightarrow{\mathcal{D}_{G_1}}.$$

By definition of G_1, $\overleftrightarrow{\mathcal{D}_{G_1}} \subset \overleftrightarrow{\mathcal{D}_E}$. Hence $\frac{\partial}{\partial(x,y)}(G) \subset \mathcal{E}$.

(iii) if G belongs to $\overleftrightarrow{\mathcal{D}_E}$, so do its derived terms.

(c) Finally, since $\mathrm{Card}(\mathcal{E}) = \mathrm{Card}(\overleftarrow{\mathcal{D}_E}) + (k-1) \times \mathrm{Card}(\overleftrightarrow{\mathcal{D}_E}) + \mathrm{Card}(\overleftrightarrow{\mathcal{D}_E})$, the proposition holds.

(3) The proof is similar as for case **(2)**, with $\overrightarrow{\mathcal{D}_E}$ playing the role of $\overleftarrow{\mathcal{D}_E}$.

□

The *index* of a hairpin expression E is the integer $\mathrm{index}(E)$ inductively defined by:

$$\mathrm{index}(F) = 0,$$

$$\text{index}(\overleftarrow{\text{H}_k}(F)) = k,\ \text{index}(\overrightarrow{\text{H}_k}(F)) = k,\ \text{index}(\text{H}'_k(F)) = k,$$
$$\text{index}(G_1 + G_2) = \max(\text{index}(G_1), \text{index}(G_2)),$$

where H is any anti-morphism over Γ^*, $k > 0$ is any integer, F is any regular expression over Γ, and G_1 and G_2 are any two hairpin expressions over Γ.

Proposition 9. *Let E be a hairpin expression over an alphabet Γ. Then $\overleftrightarrow{\mathcal{D}_E}$ is a finite set the cardinal of which is upper bounded by $k \times (\frac{2m(m+1)(m+2)}{3} - 3) + n$, where k is the index of E, and $m = n + h$ with n its width and h its star number.*

Proof. Directly deduced from Proposition 7 and from Proposition 8 for the non-sum cases. Whenever $E = G_1 + G_2$, let us set for $i \in \{1, 2\}$, n_i the width of G_i, h_i its star number, k_i its index and $m_i = n_i + h_i$. Without loss of generality suppose that $k_1 \geq k_2$. Let ϕ be the function defined by:

$$\phi(k) = \begin{cases} 0 & \text{if } k = 0, \\ \frac{2k(k+1)(k+2)}{3} - 3 & \text{otherwise.} \end{cases}$$

It can be checked that the following proposition **P** holds:

$$\phi(k_1 + k_2) \geq \phi(k_1) + \phi(k_2).$$

By induction and from **P** it holds:

$$\begin{aligned} \text{Card}(\overleftrightarrow{\mathcal{D}_{G_1}}) + \text{Card}(\overleftrightarrow{\mathcal{D}_{G_2}}) &\leq k_1 \times \phi(m_1) + n_1 + k_2\phi(m_2) + n_2 \\ &\leq k_1 \times (\phi(m_1) + \phi(m_2)) + n \\ &\leq k_1 \times \phi(m_1 + m_2) + n \end{aligned}$$

□

This finite set of two-sided derived terms allows us to extend the finite derived term automaton to hairpin expressions.

Definition 7. *Let E be a hairpin expression over an alphabet Γ. Let $A = (\Sigma_\Gamma, Q, I, F, \delta)$ be the NFA defined by:*

- $Q = \{E\} \cup \overleftrightarrow{\mathcal{D}_E}$,
- $I = \{E\}$,
- $F = \{E' \in Q \mid \varepsilon \in L(E')\}$,
- $\forall (x, y) \in \Sigma_\Gamma, \forall E' \in Q,\ \delta(E', (x, y)) = \frac{\partial}{\partial_{(x,y)}}(E')$.

The automaton A is the two-sided derived term automaton *of E.*

By construction, A is a Γ-couple NFA where Γ is the alphabet of E.

Example 3. *Let E be the hairpin expression of Example 2. The derived term automaton of E is the automaton presented in Figure 3.*

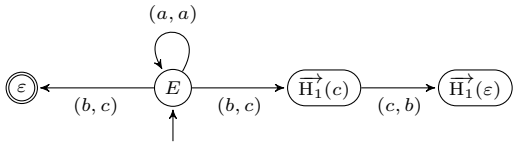

Figure 3: The Derived Term Automaton of the Expression E.

Proposition 10. *Let E be a hairpin expression over an alphabet Γ and A be the two-sided derived term automaton of E. Then: $L(E) = L_\Gamma(A)$.*

Proof. Let $A = (\Sigma, Q, I, F, \delta)$, let w be a word in Γ^* and let E' be a state in Q. Let us show that the following proposition **(P)** is satisfied: $w \in \overrightarrow{L}_\Gamma(E') \Leftrightarrow w \in L(E')$. By recurrence over the length of w.

(I) If $w = \varepsilon$, then:

$w \in \overrightarrow{L}_\Gamma(E')$

$\Leftrightarrow E' \in F$ (Lemma 3)

$\Leftrightarrow \varepsilon \in L(E')$ (Construction of A)

$\Leftrightarrow w \in L(E')$.

(II) Let us suppose that $|w| > 0$. Then $\exists (x, y) \in \Sigma_\Gamma$, $\exists w' \in \Gamma^*$ such that $w = xw'y$.

(a) If E' is a simple hairpin expression, then

$xw'y \in \overrightarrow{L}_\Gamma(E')$

$\Leftrightarrow w' \in (x, y)^{-1}(\overrightarrow{L}_\Gamma(E'))$ (Definition 1)

$\Leftrightarrow w' \in \bigcup_{E'' \in \delta(E', (x,y))} \overrightarrow{L}_\Gamma(E'')$ (Corollary 2)

$\Leftrightarrow w' \in \bigcup_{E'' \in \frac{\partial}{\partial(x,y)}(E')} \overrightarrow{L}_\Gamma(E'')$ (Construction of A)

$\Leftrightarrow w' \in \bigcup_{E'' \in \frac{\partial}{\partial(x,y)}(E')} L(E'')$ (Recurrence hypothesis)

$\Leftrightarrow w' \in (x, y)^{-1}(L(E'))$ (Proposition 6)

$\Leftrightarrow xw'y \in L(E')$ (Definition 1) $\Leftrightarrow w \in L(E')$.

(b) If $E' \in \{\overleftarrow{\mathrm{H}}_k(F), \overrightarrow{\mathrm{H}}_k(F), \mathrm{H}'_k(F)\}$, then it holds $w \in L(E') \Rightarrow y = \mathrm{H}(x)$ (according to Lemma 5). Consequently, if $y \neq \mathrm{H}(x)$,$\delta(E', (x, y)) = \emptyset$ and $w \notin \overrightarrow{L}_\Gamma(E')$. Hence, since $w \notin L(E')$, proposition is satisfied. Let us now suppose that $y = \mathrm{H}(x)$. Since $(\varepsilon, \varepsilon) \notin \Sigma_\Gamma$, $(x, y) \in \Gamma \times \Gamma$.

$xw'\mathrm{H}(x) \in \overrightarrow{L}_\Gamma(E')$

$\Leftrightarrow w' \in (x, \mathrm{H}(x))^{-1}(\overrightarrow{L}_\Gamma(E'))$ (Definition 1)

$\Leftrightarrow w' \in \bigcup_{E'' \in \delta(E', (x,\mathrm{H}(x)))} \overrightarrow{L}_\Gamma(E'')$ (Corollary 2)

$\Leftrightarrow w' \in \bigcup_{E'' \in \frac{\partial}{\partial(x,\mathrm{H}(x))}(E')} \overrightarrow{L}_\Gamma(E'')$ (Construction of A)

$\Leftrightarrow w' \in \bigcup_{E'' \in \frac{\partial}{\partial(x,\mathrm{H}(x))}(E')} L(E'')$ (Recurrence hypothesis)

$\Leftrightarrow w' \in (x, \mathrm{H}(x))^{-1}(L(E''))$ (Proposition 6)

$\Leftrightarrow xw'\mathrm{H}(x) \in L(E')$ (Definition 1)

$\Leftrightarrow w \in L(E')$

Finally,

$L_\Gamma(A) = \bigcup_{i \in I} \overrightarrow{L}_\Gamma(i)$ (Lemma 2)

$= \overrightarrow{L}_\Gamma(E)$ (Construction of A)

$= L(E)$ (proposition **P**). □

Theorem 2. *Let A be the two-sided derived term automaton of a hairpin expression E over an alphabet Γ and let k be the index of E. Then $L_\Gamma(A) = L(E)$. Furthermore A has at most $k \times (\frac{2m \times (m+1) \times (m+2)}{3} - 3) + n + 1$ states where $m = n + h$, with n the width of E and h its star number.*

Proof. Corollary of Proposition 10 and of Proposition 9. □

Finally, the computation of the two-sided derived term automaton provides an alternative proof of the following theorem.

Theorem 3. *The language denoted by a hairpin expression is linear context-free.*

Proof. According to Theorem 1 and to Proposition 10. □

6 The $(\mathrm{H}, 0)$-Completion

In the literature, the case where $k = 0$ is usually not considered. Nevertheless, this case is interesting since the associated derivation computation yields a recognizer with a linear number of states w.r.t. the width of the expression.

Let L_1 and L_2 be two languages over an alphabet Γ and H be an anti-morphism over Γ^*. The $(\mathrm{H}, 0)$-*completion* of L_1 and L_2 is the language $\mathrm{H}_0(L_1, L_2) = \{\alpha\gamma\mathrm{H}(\alpha) \mid \alpha, \gamma \in \Gamma^* \ \wedge \ (\alpha\gamma \in L_1 \ \vee \ \gamma\mathrm{H}(\alpha) \in L_2)\}$. As in the general case, the $(\mathrm{H}, 0)$-completion can be defined as the union of two unary operators $\overleftarrow{\mathrm{H}_0}$ and $\overrightarrow{\mathrm{H}_0}$.

The *left* (resp. *right*) $(\mathrm{H}, 0)$-*completion* of a language L over an alphabet Γ is the language $\overleftarrow{\mathrm{H}}_0(L) = \{\alpha\gamma\mathrm{H}(\alpha) \mid \alpha, \gamma \in \Gamma^* \ \wedge \ \gamma\mathrm{H}(\alpha) \in L\}$ (resp. $\overrightarrow{\mathrm{H}}_0(L) = \{\alpha\gamma\mathrm{H}(\alpha) \mid \alpha, \gamma \in \Gamma^* \ \wedge \ \alpha\gamma \in L\}$).

Let E be a regular expression over Γ and H be an anti-morphism over Γ^*. The *left* (resp. *right*) $(\mathrm{H}, 0)$-*completion* of E is the expression $\overleftarrow{\mathrm{H}}_0(E)$ (resp. $\overrightarrow{\mathrm{H}}_0(E)$) that denotes $\overleftarrow{\mathrm{H}}_0(L(E))$ (resp. $\overrightarrow{\mathrm{H}}_0(L(E))$).

Lemma 9. *Let Γ be an alphabet and* H *be an anti-morphism over Γ^*. Let L be a language over Γ. Then the two following conditions are satisfied:*

- $\varepsilon \in \overrightarrow{\mathrm{H}}_0(L) \Leftrightarrow \varepsilon \in L$,
- $\varepsilon \in \overleftarrow{\mathrm{H}}_0(L) \Leftrightarrow \varepsilon \in L$.

Proof. Trivially proved from the definitions of left and right $(\mathrm{H}, 0)$-completions. □

We now consider the construction of a recognizer for the $(\mathrm{H}, 0)$-completion of a regular expression E. On the opposite of the general case, it is not necessary to consider the whole computation of partial derivatives. We show that it is sufficient to consider one-sided partial derivatives of regular expression.

Definition 8. *Let Γ be an alphabet and* H *be an anti-morphism over Γ^*. Let F be a regular expression over Γ. Let $E = \overrightarrow{\mathrm{H}}_0(F)$ (resp. $E = \overleftarrow{\mathrm{H}}_0(F)$). The* effective subset associated with E *is the set defined by:*

$$\mathcal{E} = \overrightarrow{\mathrm{H}}_0(\overleftarrow{\mathcal{D}_F}) \cup \overleftarrow{\mathcal{D}_F},$$
$$\textit{(resp. } \mathcal{E} = \overleftarrow{\mathrm{H}}_0(\overrightarrow{\mathcal{D}_F}) \cup \overrightarrow{\mathcal{D}_F}\textit{).}$$

Definition 9. *Let* Γ *be an alphabet and* H *be an anti-morphism over* Γ^*. *Let* F *be a regular expression over* Γ. *Let* $E = \overrightarrow{\mathrm{H}}_0(F)$ *(resp.* $E = \overleftarrow{\mathrm{H}}_0(F)$*). Let* $\mathcal{E}$ *be the effective subset associated with* E. *Let* $A = (\Sigma_\Gamma, Q, I, F, \delta)$ *be the couple NFA defined by:* $Q = \{E\} \cup \mathcal{E}$, $I = \{E\}$, $F = \{E' \in Q \mid \varepsilon \in L(E')\}$, $\forall (x,y) \in \Sigma_\Gamma, \forall E' \in Q$,

$$\delta(E',(x,y)) = \begin{cases} \overrightarrow{\mathrm{H}}_0(\frac{\partial}{\partial_x}(E'')) & \text{if } y = \mathrm{H}(x) \ \wedge \ E' = \overrightarrow{\mathrm{H}}_0(E''), \\ \frac{\partial}{\partial_x}(E'') & \text{if } y = \varepsilon \ \wedge \ E' = \overrightarrow{\mathrm{H}}_0(E''), \\ \frac{\partial}{\partial_x}(E') & \text{if } y = \varepsilon \ \wedge \ E' \text{ is a regular expression}, \\ \emptyset & \text{otherwise}, \end{cases}$$

$$\textit{resp. } \delta(E',(x,y)) = \begin{cases} \overleftarrow{\mathrm{H}}_0((E'')\frac{\partial}{\partial_y}) & \text{if } y = \mathrm{H}(x) \ \wedge \ E' = \overleftarrow{\mathrm{H}}_0(E''), \\ (E'')\frac{\partial}{\partial_y} & \text{if } x = \varepsilon \ \wedge \ E' = \overleftarrow{\mathrm{H}}_0(E''), \\ (E')\frac{\partial}{\partial_y} & \text{if } x = \varepsilon \ \wedge \ E' \text{ is a regular expression}, \\ \emptyset & \text{otherwise}. \end{cases}$$

The automaton A *is said to be the* effective automaton *of* E.

Theorem 4. *Let* F *be a regular expression over an alphabet* Γ. *Let* A *be the effective automaton of the expression* $E = \overrightarrow{\mathrm{H}}_0(F)$ *(resp.* $E = \overleftarrow{\mathrm{H}}_0(F)$*). Then* $L_\Gamma(A) = L(E)$. *Furthermore* A *has at most* $2n+1$ *states where* n *is the width of* E.

Proof. Let us set $A = (\Sigma_\Gamma, Q, I, F, \delta)$.

(I) Let us show now that $L_\Gamma(A) = L(E)$.

(a) Let us suppose that $E = \overrightarrow{\mathrm{H}}_0(F)$. Let w be a word in Γ^*. Let us show by recurrence over the length of w that for any state E' in Q, $w \in L(E') \Leftrightarrow w \in \overrightarrow{L}_\Gamma(E')$.

(1) If $w = \varepsilon$, $w \in L(E') \Leftrightarrow E' \in F \Leftrightarrow w \in \overrightarrow{L}_\Gamma(E')$.

(2) Let w be a word different from ε.

(i) If E' is a regular expression, $a^{-1}(L(E')) = \bigcup_{E'' \in \frac{\partial}{\partial_a}(E')} L(E'')$. Hence since there exists a in Γ and w' in Γ^* such that $w = aw'$, it holds:

$aw' \in L(E') \Leftrightarrow w' \in a^{-1}(L(E')) \Leftrightarrow w' \in \bigcup_{E'' \in \frac{\partial}{\partial_a}(E')} L(E'')$

$\Leftrightarrow w' \in \bigcup_{E'' \in \frac{\partial}{\partial_a}(E')} \overrightarrow{L}_\Gamma(E'')$

$\Leftrightarrow w' \in \bigcup_{E'' \in \delta(E',(a,\varepsilon))} \overrightarrow{L}_\Gamma(E'')$ (Recurrence Hypothesis)

$\Leftrightarrow aw' \in \overrightarrow{L}_\Gamma(E')$.

(ii) If $E' = \overleftarrow{\mathrm{H}}_0(E'')$ then:

$w \in L(\overleftarrow{\mathrm{H}}_0(E'')) \Leftrightarrow \exists \alpha, \gamma \in \Gamma^*, (w = \alpha\gamma\mathrm{H}(\alpha) \ \wedge \ \alpha\gamma \in L(E''))$

$\Leftrightarrow \exists a \in \Gamma, \gamma \in \Gamma^*, \alpha' \in \Gamma^*, ((w = \gamma \ \wedge \ \gamma \in L(E'')) \ \vee \ (w = a\alpha'\gamma\mathrm{H}(\alpha')\mathrm{H}(a) \ \wedge \ a\alpha'\gamma \in L(E''))$

$\Leftrightarrow \exists a \in \Gamma, \gamma \in \Gamma^*, \alpha' \in \Gamma^*, w' \in \Gamma^*, ((w = aw' \ \wedge \ w' \in a^{-1}(L(E''))) \ \vee \ (w = a\alpha'\gamma\mathrm{H}(\alpha')\mathrm{H}(a) \ \wedge \ \alpha'\gamma \in a^{-1}(L(E''))))$

$\Leftrightarrow \exists a \in \Gamma, \gamma \in \Gamma^*, \alpha' \in \Gamma^*, w' \in \Gamma^*, ((w = aw' \ \wedge \ w' \in \bigcup_{E'' \in \frac{\partial}{\partial_a}(E')} L(E'')) \ \vee \ (w = a\alpha'\gamma\mathrm{H}(\alpha')\mathrm{H}(a) \ \wedge \ \alpha'\gamma \in \bigcup_{E'' \in \frac{\partial}{\partial_a}(E')} L(E'')))$

$\Leftrightarrow \exists a \in \Gamma, \gamma \in \Gamma^*, \alpha' \in \Gamma^*, w' \in \Gamma^*, ((w = aw' \wedge w' \in \bigcup_{E'' \in \frac{\partial}{\partial_a}(E')} \overrightarrow{L}_\Gamma(E'')) \vee$
$(w = a\alpha'\gamma \mathrm{H}(\alpha')\mathrm{H}(a) \ \wedge \ \alpha'\gamma \in \bigcup_{E'' \in \frac{\partial}{\partial_a}(E')} \overrightarrow{L}_\Gamma(E'')))$ (Recurrence Hypothesis)

$\Leftrightarrow \exists a \in \Gamma, \gamma \in \Gamma^*, \alpha' \in \Gamma^*, w' \in \Gamma^*, ((w = aw' \wedge w' \in \bigcup_{E'' \in \delta(E',(a,\varepsilon))} \overrightarrow{L}_\Gamma(E'')) \vee$
$(w = a\alpha'\gamma \mathrm{H}(\alpha')\mathrm{H}(a) \ \wedge \ \alpha'\gamma \in \bigcup_{E'' \in \delta(E',(a,\mathrm{H}(a)))} \overrightarrow{L}_\Gamma(E'')))$

$\Leftrightarrow w \in \overrightarrow{L}_\Gamma(E')$.

Finally since $L(A) = \overrightarrow{L}_\Gamma(E)$ and since $L(E) = \overrightarrow{L}_\Gamma(E)$, then $L(A) = L(E)$.

(b) The case where $E = \overleftarrow{\mathrm{H}}_0(F)$ is based on the same reasoning.

(II) Let $\mathcal{E} = \overrightarrow{\mathrm{H}}_0(\overleftarrow{\mathcal{D}_F}) \cup \overleftarrow{\mathcal{D}_F}$ be the effective subset associated with E (resp. $\mathcal{E} = \overleftarrow{\mathrm{H}}_0(\overrightarrow{\mathcal{D}_F}) \cup \overrightarrow{\mathcal{D}_F}$). Since $\mathrm{Card}(\overleftarrow{\mathcal{D}_F}) \leq n$ (resp. $\mathrm{Card}(\overrightarrow{\mathcal{D}_F}) \leq n$), the number of states of A is at most $2n$. Finally, since $Q = \mathcal{E} \cup \{E\}$, it holds that A has at most $2n + 1$ states. □

Example 4. *Let* H *be the anti-morphism defined in Example 2. Let* $E = \overrightarrow{\mathrm{H}_0}(a^*bc)$. *Notice that* $\overleftarrow{\mathcal{D}}_{a^*bc} = \{a^*bc, c, \varepsilon\}$. *Hence the effective subset associated with* E *is the set* $\{\overrightarrow{\mathrm{H}_0}(a^*bc), \overrightarrow{\mathrm{H}_0}(c), \overrightarrow{\mathrm{H}_0}(\varepsilon), a^*bc, c, \varepsilon\}$.

The effective automaton A *of* E *is given Figure 4.*

It can be checked that $L(A) = \{a^nbc \mid n \in \mathbb{N}\} \cup \{a^nbca^n \mid n \in \mathbb{N}\} \cup \{a^nbcca^n \mid n \in \mathbb{N}\} \cup \{a^nbcbca^n \mid n \in \mathbb{N}\}$ *that is exactly* $L(E)$ *(see Table 1).*

α	γ	$\mathrm{H}(\alpha)$
ε	a^nbc	ε
a^n	bc	a^n
a^nb	c	ca^n
a^nbc	ε	bca^n

Table 1: The Language $L(E)$

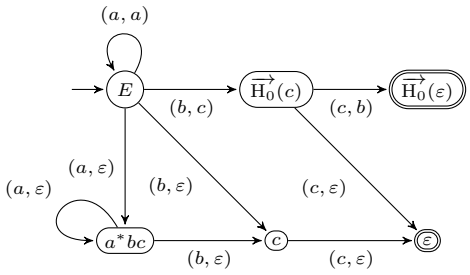

Figure 4: The Effective Automaton of the Expression E

7 Conclusion

This paper provides an alternative proof of the fact that hairpin completions of regular languages are linear context-free. This proof is obtained by considering the family of regular expressions extended to hairpin operators and by computing their partial derivatives, a technique that has already been applied to regular expressions extended to boolean operators [4], to multi-tilde-bar operators [5] and to approximate operators [8]. Moreover it is a constructive proof since it is based on the computation of a polynomial size recognizer for hairpin completions of regular languages. We also proved that it is possible to compute a linear size recognizer for $(H, 0)$-completions of regular languages.

References

[1] V. Antimirov. Partial derivatives of regular expressions and finite automaton constructions. *Theoret. Comput. Sci.*, 155:291–319, 1996.

[2] P. Bottoni, A. Labella, V. Manca, and V. Mitrana. Superposition based on watson-crick-like complementarity. *Theory Comput. Syst.*, 39(4):503–524, 2006.

[3] J. A. Brzozowski. Regular-like expressions for some irregular languages. In *SWAT (FOCS)*, pages 278–286. IEEE Computer Society, 1968.

[4] P. Caron, J.-M. Champarnaud, and L. Mignot. Partial derivatives of an extended regular expression. In Adrian Horia Dediu, Shunsuke Inenaga, and Carlos Martín-Vide, editors, *LATA*, volume 6638 of *Lecture Notes in Computer Science*, pages 179–191. Springer, 2011.

[5] P. Caron, J.-M. Champarnaud, and L. Mignot. Multi-tilde-bar derivatives. In Nelma Moreira and Rogério Reis, editors, *CIAA*, volume 7381 of *Lecture Notes in Computer Science*, pages 321–328. Springer, 2012.

[6] J. Castellanos and V. Mitrana. Some remarks on hairpin and loop languages. In Masami Ito, Gheorghe Paun, and Sheng Yu, editors, *Words, Semigroups, and Transductions*, pages 47–58. World Scientific, 2001.

[7] J.-M. Champarnaud, J.-P. Dubernard, H. Jeanne, and L. Mignot. Two-sided derivatives for regular expressions and for hairpin expressions. In *LATA*, 2013. To appear.

[8] J.-M. Champarnaud, H. Jeanne, and L. Mignot. Approximate regular expressions and their derivatives. In Adrian Horia Dediu and Carlos Martín-Vide, editors, *LATA*, volume 7183 of *Lecture Notes in Computer Science*, pages 179–191. Springer, 2012.

[9] D. Cheptea, C. Martìn-Vide, and V. Mitrana. A new operation on words suggested by DNA biochemistry: hairpin completion. *Transgressive Computing*, pages 216–228, 2006.

[10] V. Diekert, S. Kopecki, and V. Mitrana. On the hairpin completion of regular languages. In Martin Leucker and Carroll Morgan, editors, *ICTAC*, volume 5684 of *Lecture Notes in Computer Science*, pages 170–184. Springer, 2009.

[11] V. Diekert, S. Kopecki, and V. Mitrana. Deciding regularity of hairpin completions of regular languages in polynomial time. *Inf. Comput.*, 217:12–30, 2012.

[12] M. Ito, P. Leupold, F. Manea, and V. Mitrana. Bounded hairpin completion. *Inf. Comput.*, 209(3):471–485, 2011.

[13] L. Kari, S. Kopecki, and S. Seki. Iterated hairpin completions of non-crossing words. In Mária Bieliková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán, editors, *SOFSEM*, volume 7147 of *Lecture Notes in Computer Science*, pages 337–348. Springer, 2012.

[14] L. Kari, S. Seki, and S. Kopecki. On the regularity of iterated hairpin completion of a single word. *Fundam. Inform.*, 110(1-4):201–215, 2011.

[15] S. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, Ann. Math. Studies 34:3–41, 1956. Princeton U. Press.

[16] S. Kopecki. On iterated hairpin completion. *Theor. Comput. Sci.*, 412(29):3629–3638, 2011.

[17] S. Lombardy and J. Sakarovitch. Derivatives of rational expressions with multiplicity. *Theor. Comput. Sci.*, 332(1-3):141–177, 2005.

[18] F. Manea, C. Martín-Vide, and V. Mitrana. On some algorithmic problems regarding the hairpin completion. *Discrete Applied Mathematics*, 157(9):2143–2152, 2009.

[19] F. Manea, C. Martín-Vide, and V. Mitrana. Hairpin lengthening. In Fernando Ferreira, Benedikt Löwe, Elvira Mayordomo, and Luís Mendes Gomes, editors, *CiE*, volume 6158 of *Lecture Notes in Computer Science*, pages 296–306. Springer, 2010.

[20] F. Manea and V. Mitrana. Hairpin completion versus hairpin reduction. In S. Barry Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *CiE*, volume 4497 of *Lecture Notes in Computer Science*, pages 532–541. Springer, 2007.

[21] F. Manea, V. Mitrana, and T. Yokomori. Two complementary operations inspired by the DNA hairpin formation: Completion and reduction. *Theor. Comput. Sci.*, 410(4-5):417–425, 2009.

[22] F. Manea, V. Mitrana, and T. Yokomori. Some remarks on the hairpin completion. *Int. J. Found. Comput. Sci.*, 21(5):859–872, 2010.

[23] V. Mitrana, F. Manea, and C. Martín-Vide. On some algorithmic problems regarding the hairpin completion. *Electronic Notes in Discrete Mathematics*, 27:71–72, 2006.

[24] J. M. Sempere. On a class of regular-like expressions for linear languages. *Journal of Automata, Languages and Combinatorics*, 5(3):343–354, 2000.